

AD-A055 538

NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF

F/G 21/5

STRESS ANALYSIS OF CERAMIC GAS TURBINE BLADES BY THE FINITE ELE--ETC(U)

MAR 78 J H PREISEL

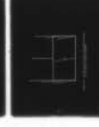
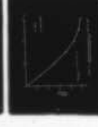
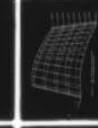
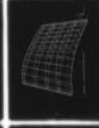
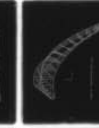
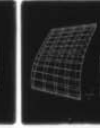
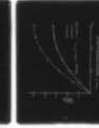
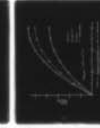
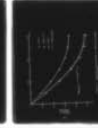
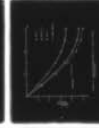
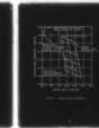
UNCLASSIFIED

NPS-69-78-010-PT-2

NL

1 of 2

AD
A055 538



FOR FURTHER TRANSMISSION

2

AD A 055538

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DDC
JUN 22 1978
F

AD No. ~~1~~
DDC FILE COPY

THESIS

STRESS ANALYSIS OF CERAMIC GAS TURBINE BLADES
BY THE FINITE ELEMENT METHOD
PART II

by

John H. Preisel, Jr.

March 1978

Thesis Advisor:

Gilles Cantin

Approved for public release; distribution unlimited.

Prepared for: Naval Sea Systems Command
Washington, D.C.
Code 0331

78 06 15 091

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 NPS69-78-010-PT-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Stress Analysis of Ceramic Gas Turbine Blades by the Finite Element Method, Part II.		5. TYPE OF REPORT & PERIOD COVERED Engineering Thesis; March 1978
7. AUTHOR(s) 10 John H. Preisel, Jr		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N0002478WR8G192
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940 (12) 130p.		12. REPORT DATE 11 March 1978
		13. NUMBER OF PAGES 132
		14. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. 9 Master's thesis,		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Gas Turbine, Stress Analysis, Finite Element; Pre- and Post-Processor		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The complex geometry of many parts of a ceramic gas turbine engine and the need to know the stresses in these parts to a high degree of accuracy indicate that the finite element method should be employed. One analysis code used a finite element program based on an isoparametric 12-node brick. This allowed quadratic variation in one coordinate direction, but only linear variation in the other two coordinate directions. This intro-		

DD FORM 1473
1 JAN 73
(Page 1)

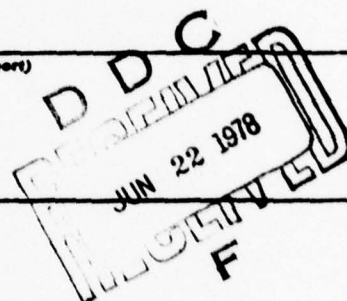
EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251 450

1

LB



duced an artificial stiffness into the structure. This study was made to determine the effects of using a totally quadratic isoparametric brick, vice one which is quadratic in only one direction.

A distributed load preprocessor was also developed and tested.

FORTTRAN IV was used throughout.

ACCESSION for	
NTIS	Write Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
RESTRICTION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
SPECIAL	
A	

Approved for public release; distribution unlimited.

Stress Analysis of Ceramic Gas Turbine Blades
by the Finite Element Method
Part II

by

John H. Preisel, Jr.
Lieutenant, United States Navy
B.S.Mar.E., United States Naval Academy, 1972

Submitted in partial fulfillment of the
requirements for the degrees of

MECHANICAL ENGINEER

and

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1978

Author

John H. Preisel, Jr.

Approved by:

Gilles Bantier

Thesis Advisor

R. E. Newton

Second Reader

Allen E. Fuhs

Chairman, Department of Mechanical Engineering

George J. Halpin

Dean of Science and Engineering

78 06 15 09 1

ABSTRACT

The complex geometry of many parts of a ceramic gas turbine engine and the need to know the stresses in these parts to a high degree of accuracy indicate that the finite element method should be employed. One analysis code used a finite element program based on an isoparametric 12-node brick. This allowed quadratic variation in one coordinate direction, but only linear variation in the other two coordinate directions. This introduced an artificial stiffness into the structure. This study was made to determine the effects of using a totally quadratic isoparametric brick, vice one which is quadratic in only one direction.

A distributed load preprocessor was also developed and tested.

FORTRAN IV was used throughout.

TABLE OF CONTENTS

I.	INTRODUCTION - - - - -	12
	A. WHY CERAMICS? - - - - -	12
	B. DESIGN CONSIDERATIONS - - - - -	13
	C. PROJECT BACKGROUND - - - - -	13
	D. REASON FOR THESIS - - - - -	14
II.	THEORY - - - - -	15
III.	BENDING SPECIMEN PROBLEM - - - - -	18
	A. FOUR POINT FLEXURE TESTS - - - - -	18
	B. MODELING TECHNIQUES - - - - -	18
	C. FLEXURE CASE STUDY - - - - -	19
	1. Analysis of Results - - - - -	19
	D. TEMPERATURE GRADIENT CASE STUDIES - - - - -	22
	1. Results of Case Study Using Poisson's Ratio (ν) - - - - -	23
	2. Results of Case Study with $\nu = 0$ - - - - -	25
IV.	AIRFOIL MODELING - - - - -	26
	A. DEVELOPMENT OF AIRFOIL PROFILES - - - - -	26
	B. THE NEGATIVE JACOBIAN DETERMINANT - - - - -	27
	C. MESH VERIFICATION THROUGH PSAP1 - - - - -	29
V.	AIRFOIL CASE STUDY - - - - -	32
	A. CENTRIFUGAL LOAD - - - - -	32
	B. THERMAL LOAD - - - - -	33
	C. CONCLUSIONS FROM AIRFOIL STUDY - - - - -	34

VI.	DISTRIBUTED LOAD PREPROCESSOR - - - - -	35
A.	DEVELOPMENT OF THEORY - - - - -	35
B.	PROGRAM STRUCTURE - - - - -	38
C.	PROGRAM VERIFICATION - - - - -	39
D.	USER'S MANUAL - - - - -	40
E.	USE OF PRESSURE LOAD PREPROCESSOR ON CP/CMS - - - - -	42
VII.	USING PSAP1 AS A DISPLACEMENT POSTPROCESSOR - -	43
A.	MODIFICATION OF ADINA - - - - -	43
B.	MODIFICATION OF PSAP1 - - - - -	44
C.	USER'S MANUAL FOR DISPLACEMENT POSTPROCESSING - - - - -	44
VIII.	CONCLUSIONS AND RECOMMENDATIONS - - - - -	47
A.	CONCLUSIONS - - - - -	47
B.	RECOMMENDATIONS - - - - -	47
IX.	FIGURES - - - - -	49
X.	TABLES - - - - -	88
APPENDIX A:	PREPARATION OF TEMPERATURE TAPE - - - - -	93
APPENDIX B:	NODAL TEMPERATURE TAPE PROGRAM - - - - -	96
APPENDIX C:	SAMPLE INPUT DECK AND RESULTS FOR DISTRIBUTED LOAD PREPROCESSOR - - - - -	103
APPENDIX D:	LISTING OF DISTRIBUTED LOAD PREPROCESSOR - - - - -	108
LIST OF REFERENCES	- - - - -	128
INITIAL DISTRIBUTION LIST	- - - - -	130

LIST OF TABLES

I.	Mesh Characteristics - - - - -	88
II.	Comparison of Maximum Stresses in Airfoil Root Section - - - - -	89
III.	Comparison of Maximum Principal Stresses in Airfoil Root Section - - - - -	90
IV.	Comparison of Selected Thermal Stresses in Airfoil Root Section - - - - -	91
V.	Face Numbering Convention for Pressure Loads - - - - -	92

LIST OF FIGURES

1.	Engine Design Parameters - - - - -	49
2.	Ceramic Hot-Section Components - - - - -	50
3.	Two-Dimensional Plane Elements - - - - -	51
4.	Schematic of 4-Point Flexure Rig - - - - -	52
5a.	Bend Specimen Free-Body Diagram - - - - -	53
5b.	Bend Specimen Shear Diagram - - - - -	53
5c.	Bend Specimen Moment Diagram - - - - -	53
6a.	Four-Point Bend Specimen Model with Midspan Boundary Condition - - - - -	54
6b.	x-z View of Half-Span with Guided End Condition - - - - -	55
6c.	y-z View of Specimen with Guided Midplane Boundary Condition - - - - -	55
7.	Cross-Sectional View of Deformed Specimen Showing x-z Plane of Symmetry - - - - -	56
8a.	12-Node Brick Orientation for Case 1 - - - - -	57
8b.	12-Node Brick Orientation for Case 2 - - - - -	57
8c.	12-Node Brick Orientation for Case 3 - - - - -	57
8d.	20-Node Brick; Case 4 - - - - -	57
9a.	Free Body Diagram for Flexure Case Study - - - - -	58
9b.	Free Body Diagram for Temperature Gradient Case Study - - - - -	58
10.	Normalized Displacement vs. Span for Steel Bend Specimen - - - - -	59
11.	Normalized Displacement vs. Span for Ceramic Bend Specimen - - - - -	60

12.	Normalized Displacement vs. Span for Temperature Gradient Analyses ($v=0.2$) - - - - -	-61
13.	Normalized Displacement vs. Span for Temperature Gradient Analyses ($v = 0$) - - - - -	-62
14.	Airfoil Tip Section Profile - - - - -	-63
15.	Airfoil Hub Section Profile - - - - -	-64
16.	20-Node Airfoil Mesh View 1 - - - - -	-65
17.	20-Node Airfoil Mesh View 2 - - - - -	-66
18.	20-Node Airfoil Mesh View 3 - - - - -	-67
19.	Airfoil Mesh Exploded Plot with Elements Numbered - - - - -	-68
20.	Trailing Edge Element Showing Re-entrant Angle - - - - -	-69
21.	12-Node Airfoil Section Showing Truncation of Leading Edge Element - - - - -	-70
22.	12-Node Airfoil Mesh View 1 - - - - -	-71
23.	Deformed 20-Node Airfoil Superposed on Original Geometry - View 1 - - - - -	-72
24.	Deformed 20-Node Airfoil Superposed on Original Geometry - View 2 - - - - -	-73
25.	Deformed 20-Node Airfoil Superposed on Deformed 12-Node Airfoil - View 1 - - - - -	-74
26.	Deformed 20-Node Airfoil Superposed on Deformed 12-Node Airofil - View 2 - - - - -	-75
27.	Airfoil Root Section Showing Location of Nodal Points used for Comparison in Tables II and III - - - - -	-76
28.	Airfoil Root Section Showing Location of Gauss Points used for Comparison in Table IV - - - - -	-77
29.	12-Node Airfoil Root Section Stress Contours - - - - -	-78

30.	20-Node Airfoil Root Section Stress Contours - - - - -	79
31.	Nodal Temperature Distribution Used for Thermal Analysis of Airfoil - - - - -	80
32.	Thermally Deformed 20-Node Airfoil Superposed on Original Geometry - - - - -	81
33.	Pressurized Element - - - - -	82
34.	Pressure Verification Problem - - - - -	83
35.	Normalized Displacement vs. Span for Bar Problem with Pressure Load - - - - -	84
36.	Hexahedral Element in Natural Coordinates - - -	85
37.	Deformed Structure for Flexure Problem - - - - -	86
38.	Postprocessing Using Displacement Vectors at the Nodes (Flexure Problem) - - - - -	87

ACKNOWLEDGEMENT

I would like to thank all of the people who assisted me in my pursuit of knowledge at NPS.

In particular, I would like to thank Professor Gilles Cantin for introducing me to the finite element method, and his guidance throughout our association.

The cooperation of the AiResearch Division of the Garrett Corporation must also be acknowledged. Mr. D. J. Tree, and Mr. H. S. Kington were especially helpful.

A special thanks goes to my wife Marybeth for her constant help, encouragement and companionship. Without her, this and many other projects would never have been undertaken.

I. INTRODUCTION

A. WHY CERAMICS?

There are three major advantages in using ceramics in the hot section of a gas turbine: improved engine performance, higher durability, and less dependence on expensive imported materials.

Simple Brayton cycle analysis shows that a higher turbine inlet temperature (TIT) results in a higher thermal efficiency, and lower specific fuel consumption. Blade material considerations usually limit TIT. Intricate blade cooling schemes, such as those used in the LM2500 and the FT9, use impingement and convective heat transfer and external film cooling, to lower the temperature of the turbine blade. This degrades engine performance and introduces manufacturing complexities. Additionally, the cooling holes are susceptible to plugging, which may lead to eventual failure.

Ceramics, however, can withstand higher turbine inlet temperatures without blade cooling. Ceramic components also have significantly higher resistance to hot corrosion than metal "superalloy" engines. The corrosion properties of ceramics make them better candidates to withstand the higher fuel impurities expected in the future.

At present, metal engines require expensive, imported metals such as chromium, cobalt, columbium, and nickel.

These metals are in limited supply and come from areas of potential political instability. Ceramics rely on inexpensive raw materials which are found in the United States.

B. DESIGN CONSIDERATIONS

Ceramics are brittle materials. This does not mean that they are necessarily fragile. A brittle material will fracture with little or no plastic deformation. A fragile material will fracture at low stress with little or no plastic deformation. It has been suggested that the word "inductile" be used instead of brittle [1]. The inductile nature of ceramics is the crux of the design problem with this material. Local stress concentrators, such as the tip of a crack, will not yield in plastic deformation. If the stress concentration is not relieved, material fracture may result.

Another major problem is the variability of data on the material properties of ceramics.

These two problems have a major impact on the design. Design procedures that were valid for most engineering metals may fail when applied to ceramics.

C. PROJECT BACKGROUND

Some of the calculations performed later in this work are concerned with the design of a ceramic engine for a Defense Advanced Research Projects Agency (DARPA) project. The prime contractor, the AiResearch Division of the Garrett Corporation, used their T76 model metal engine as

the basis for the ceramic design. They indicated that by replacing certain metal hot section parts, they could improve engine output power by 40%, and reduce specific fuel consumption by 10%, as shown in Fig. 1. An engine cross-section (Fig. 2) shows the ceramic hot section.

As mentioned earlier, the brittle nature of ceramics drives the design. In the case of the Garrett engine, a compliant layer was added in the contact region of the dovetail as an attempt to reduce the contact stresses. Blade tip clearance was increased to reduce the possibility of impact of the blade on the shroud. In addition, a probabilistic approach, based on the Weibull distribution, was used as part of a design methodology that was aimed at predicting probability of success. These are but a few of the examples of how ceramics affected the design.

D. REASON FOR THESIS

The complex geometry of many of the ceramic engine's parts, and the need to know the stresses to a high degree of accuracy, indicated that the finite element method should be employed. One analysis code used a finite element program based on an isoparametric 12-node brick. This allowed quadratic variation in one coordinate direction, but only linear variation in the other two coordinate directions. This introduced an artificial stiffness into the structure. This study was made to determine the effects of using a totally quadratic isoparametric brick, vice one which is quadratic in only one direction.

II. THEORY

Recall from strength of materials:

$$M = EI/\rho$$

where M = moment, E = Young's modulus, I = moment of inertia, and ρ = radius of curvature.

Consider a beam made of an isotropic material, with a constant cross-section. If the beam is in pure bending, the moment is constant across the span. If Young's modulus and moment of inertia are also constant, the radius of curvature is a constant. A curve which has a constant radius of curvature is an arc of a circle.

Figure 3(a) shows a 4-node plane element. There can only be linear variation on each of the sides. If the element is deformed in pure bending, each of the sides remains straight. The top and bottom surfaces do not reflect any of the curvature expected. This is a very stiff approximation.

A 6-node plane element, shown in Figs. 3(b) and 3(c), yields a better approximation to the bending phenomenon, depending on the orientation of the one quadratic side. Orientation 1 does not reflect any of the bending curvature. Orientation 2, however, does. The finite element method would represent the deflected curve, which is defined by 3 nodes, by passing a parabola through the node points. This

can give a very close approximation to the circular arc predicted by strength of materials. It should be noted, however, that orientation 2 is constrained by the same "planes remain plane" restriction applied to slender beam theory. Thus, depending on the geometry and loading, a 6-node plane element with orientation 2 may or may not accurately reflect bending for a deep beam.

A totally quadratic element, shown in Fig. 3(d), is not constrained by orientation, geometry or loading. An 8-node plane element can give a better representation of bending.

For a given material, a stiffer element, such as the 4- or 6-node plane element, will deform less than it should. Since strain is based on displacement, and stress is based on strain, the stiffer structure will predict a lower stress than is actually present. This is a potentially dangerous design situation.

The development presented here is based on a two-dimensional plane element. Similar reasoning can be applied to three-dimensional bricks. A 12-node brick, for example, has only one quadratic side. Depending on the loading, geometry, and orientation of the quadratic side, a 12-node brick could approach the correct solution. It will always, however, be stiffer than the actual structure. If it is oriented incorrectly with respect to the applied moment, it could result in seriously incorrect answers for a coarse mesh system. In addition, since two of the coordinate

directions are stiffer than the third quadratic coordinate direction, an artificial anisotropy is introduced into the problem.

A fully quadratic brick does not have the problems mentioned above. Although the discretization process introduces some artificial stiffness, it is very slight. A fully cubic brick would even be a better representation of the actual structure.

III. BENDING SPECIMEN PROBLEM

To evaluate the accuracy of a finite element computer program, it is often useful to compare its results to experimental results for a problem that can be verified experimentally. A 4-point flexure test has been used very often for this purpose [3]. This chapter explains how the flexure test can be modeled using 12-node and 20-node bricks. The results are also discussed.

A. FOUR POINT FLEXURE TESTS

Figure 4 shows a schematic of a 4-point flexure rig. It is used to provide test results from a specimen in pure bending. Figure (5a) shows a free body diagram for the specimen. Figures (5b) and (5c) show the shear and moment diagrams. The moment diagram shows that the portion of the span between the applied forces on the top surface has no shear, and maximum constant moment, i.e., pure bending.

B. MODELING TECHNIQUES

A standard 0.125 inch x 0.250 inch x 3.0 inch bend specimen was used. The specimen was modeled using a variably-noded three-dimension brick from the ADINA [4] and SAP IV [5] finite element programs. Maximum use was made of the symmetry in the problem. Only 25% of the specimen needed to be modeled, as shown in Figs. 6a,b,c. Symmetry in the spanwise and long tranverse directions was used.

A cross-sectional view of the deformed bar, Fig. 7, shows the symmetry in the long transverse direction. The mid-span cross-section was given a guided end condition (no displacement in the x-direction). The midplane in the long transverse direction was also given a guided end condition (no displacement in the y-direction).

Since the model was very regular in shape, automatic nodal point coordinate and connectivity generation were extensively used. Both were generated in the spanwise direction to take maximum advantage of the automatic generation features of the codes employed.

C. FLEXURE CASE STUDY

A 12-node brick has one quadratic direction. The other two directions are linear. When used to model the bending specimen, three distinct orientations of the brick can occur. Figure 8 shows each of these orientations with respect to the bending moment. An analysis was performed using the SAP IV program for each of the three orientations, called Case 1, Case 2, and Case 3. The applied forces were modeled as distributed loads.

1. Analysis of Results

The three-dimensional finite elements used to solve the problem would give a solution converging to an exact three-dimensional elasticity theory solution. However, such a solution in closed form does not exist. Therefore, the results were compared to the usual strength of materials beam theory. Such a theory is not perfect and circumspection

had to be used in such a comparison. For instance, beam theory does not predict any anticlastic curvature [18], but the finite element solution shows some. However, due to the boundary conditions and symmetry, beam theory should apply to the x-z plane of symmetry with a very good accuracy. The displacements of the neutral axis of this plane from $x = 0$ to $x = L$ were plotted versus the span length. Displacements were normalized by dividing them by the mid-span ($x = 0$) displacement predicted by beam theory (δ_A). This facilitates comparisons in terms of the percentage of the maximum deflection of beam theory.

Two loadings and materials were used. The first specimen was made of steel, and had a 300 lb load. The second specimen was hot pressed silicon nitride (HPSN), and had a 180 lb load. The mechanical properties of HPSN are given in Ref. [6].

The deflection curves were calculated using [7]:

$$\delta = \delta_A + \frac{M_A x^2}{2EI} - \frac{w \langle x-a \rangle^3}{6EI}$$

where $\delta_A = - \frac{w(L-a)}{6EI} (2L^2 + 2aL - a^2)$
 $M_A = w(L-a) =$ reaction end moment
 $w =$ applied force
 $E =$ Young's modulus
 $I =$ moment of inertia

Figure 9 shows that L is half the distance between supports on the lower side of the specimen, and w is the total load applied, i.e., 300 lb or 180 lb. The symbol $\langle \rangle$ denotes a

discontinuous function, i.e., for $x < a$ the quantity $\langle x-a \rangle = 0$, for $x \geq 0$ $\langle x-a \rangle = x-a$.

The results of these analyses are shown in Figs. 10 and 11. The Case 1, 2 and 3 curves correspond to three orientations of the 12-node brick. Case 1, the orientation with the quadratic side along the axis of bending, is the closest of the three to the values predicted by beam theory. The maximum deflection of the ceramic specimen is approximately 96% of the deflection predicted by beam theory. This model is slightly stiffer than beam theory. Cases 2 and 3, which have the quadratic side perpendicular to the axis of bending, are significantly stiffer models than Case 1. The maximum deflection predicted by these cases is only 70% of the deflection predicted by beam theory.

In the displacement-based finite element method, stress and displacement are related by [8]:

$$\{e\} = [B]\{u\}$$

$$\text{and} \quad \{\tau\} = [C]\{e\} = [C][B]\{u\}$$

where $\{\tau\}$ = a vector composed of the six independent components of the stress tensor.
 $\{e\}$ = a vector composed of the six independent components of the small strain tensor.
 $[B]$ = strain-displacement transformation matrix.
 $\{u\}$ = displacement matrix.
 $[C]$ = matrix of material properties.

This formulation clearly shows that a stiffer structure (i.e., less displacement and displacement gradients) will produce less stress.

Case 4, shows results which are slightly more flexible than beam theory. Since beam theory is inherently stiff due to the approximations made during its development (i.e., negligible shear stress), the Case 4 model is closer to predicting the actual stress distribution than either Cases 1, 2, 3 or beam theory. It should also be remembered that the actual structure modeled will bend in all three directions. Clearly, the only way to fully represent such a deformation is to use a fully quadratic element. Any element that is less than fully quadratic will be too stiff.

D. TEMPERATURE GRADIENT CASE STUDIES

The temperature gradient case studies used the same models as the flexure case study. Cases 1, 2, 3 refer to the same orientation of the 12-node brick as in the flexure study. Case 4 is the 20-node brick model.

Hot pressed silicon nitride was used as the specimen material. The coefficient of thermal expansion is given in Ref. [6]. A 100°F temperature gradient was imposed across the short transverse (0.125 inch). The stress-free reference temperature was 1500°F. A variably-noded (12-21) three-dimensional brick from the ADINA finite element program was used to model the problem. Nodal temperatures were input via a temperature tape. This procedure is described in Appendix A.

As in the flexure case study, the normalized deflections of the y-z plane of symmetry were plotted versus distance along the span. Beam theory was used as the basis for comparison. The deflection predicted by beam theory is [7]:

$$\delta = \delta_A + \frac{\alpha}{2t} (T_2 - T_1) x^2$$

where: $\delta_A = -\alpha L^2 (T_2 - T_1) / 2t$
 α = coefficient of thermal expansion (inch/inch/°F)
 t = depth of beam
 $(T_2 - T_1)$ = temperature difference through depth of beam
 L = half the distance between the lower reactions

Two different case studies evolved from the temperature problem. Reference [9] states that the Bernoulli-Euler assumptions can be used in practical analyses of beams under thermal loadings. The assumptions state, in part, that the effects of lateral contraction can be ignored, i.e., Poisson's ratio (ν) may be taken as equal to zero. In the case studies that follow, Poisson's ratio was used in the first study, and set equal to zero in the second.

1. Results of the Case Study using Poisson's ration (ν)

The normalized deflection curves for this case study are shown in Fig. 12. The displacements from $x=0$ to $x=L$ are plotted. Cases 2 and 3 are stiffer than beam theory. The curves show that the maximum deflection calculated for these two cases is 75% of the maximum deflection predicted by beam theory.

The results of Case 4, which used the fully quadratic 20-node brick, show excellent agreement with simple theory.

Figure 12 shows that the results of Case 1 (the 12-node brick with quadratic side along the span) predict a deflection greater than theory. It is puzzling that the results show that the element is even more flexible than the 20-node brick. The temperature algorithm used by ADINA was checked extensively. No errors were detected. No satisfactory explanation for the 12-node brick behavior has been found. It is suspected that the inherent anisotropy of the 12-node brick may be the cause of the problem. However, extensive further examination would be necessary to resolve the dilemma.

Possibly more interesting than the displacement results are the stresses calculated. Recall that if a linear temperature gradient were imposed on a bar that had no external constraints, the bar would deform into an arc of a circle. Also there would be no stress in the bar according to a fundamental theorem of elasticity. In this case study, the bottom nodes at $x = 0.75$ inch were fixed in the z -direction. The part of the specimen to the left of this constraint (i.e., toward the free end) should, however, show no stress until one reaches the immediate vicinity of the constraint. Each of the cases that use the 12-node brick predict stresses, both normal and shear, on the order of 10^3 to 10^4 PSI in the free end. The 20-node brick, however, predicts stresses of approximately 5 PSI or less in this region. ($E=45 \times 10^6$ psi, $\nu=0.2$, $\alpha=1.7 \times 10^{-6}$ in/in/ $^{\circ}$ F).

The above discussion concerning the 12-node brick models indicates the possible uncertainty that can result from using this element in thermal calculations.

2. Results of Case Study with $\nu = 0$

This case study was undertaken in an attempt to explain the results of the Case 1 orientation of the 12-node brick discussed above. Everything was the same as in the previous study, except that the effects of Poisson's ratio were neglected. The results are shown in Fig. 13. The Case 2 and 3 models have become significantly stiffer. The maximum deflection was approximately 67% of the value predicted by theory, as compared to approximately 75% when Poisson's ratio was used. The results for Case 4 have remained the same (to three significant digits). The displacements for Case 1 have decreased to such a degree that they were almost identical to Case 4. The spurious free end stresses cited previously remained.

In summary, by ignoring the effects of Poisson's ratio, three of the models have become significantly stiffer, while the fully quadratic model has remained substantially the same and in close agreement with the theory.

It is interesting to note that if the specimen were truly unconstrained and had a linear temperature gradient, the results of analyses that take into account Poisson's ratio should agree exactly with the results of analyses that do not [9].

IV. AIRFOIL MODELING

This chapter discusses the techniques developed to construct a finite element mesh for the airfoil section of a ceramic gas turbine blade. This particular engine airfoil lent itself to detailed analysis because of the relative ease with which the mesh could be generated. This is due to the linear fairing between the tip and root sections of the airfoil. Although linear fairing was not optimal from an aerodynamic point of view, it was desirable from a manufacturing point of view. Its relative ease of fabrication caused it to be used in practice.

A. DEVELOPMENT OF AIRFOIL PROFILES

The Garrett Corporation provided Mylar cross-sectional profiles of the first stage rotor blade of their experimental ceramic turbine [10]. These profiles were for various radial distances as measured from the axis of rotation along the stacking axis of the blade. In addition to an outline of the cross-section and its relation to the stacking axis, a table of x-y coordinates was given for the profile. These coordinates described 62 points on the profile.

The 62 profile points were assembled into x and y matrices. These matrices were used to develop cross-sectional plots on the time sharing terminal and later on the Calcomp plotter. Figures 14 and 15 show representative cross-

sections of the tip and hub sections. The profile points are marked by an 'x' and are connected by straight line segments.

Since there was linear fairing between the tip and hub, it could be easily verified that the coordinates listed on the Mylar drawings were given at the same relative positions along the foil profile. In other words, the i th point on the $R=3.1$ inch profile corresponded to the i th point on the $R=4.0$ inch profile, where R is the radial distance along the stacking axis from the axis of rotation.

The root profile was then divided into elements. Care was taken to develop elements with convex corners. Also, a relatively fine mesh was desirable in the leading and trailing edge areas of the airfoil to enable accurate modeling of the anticipated stress concentrations there. The resulting leading and trailing edge elements proved quite troublesome; this will be discussed more fully below.

The mesh developed in the airfoil root section dictated the mesh in the tip section. This was due to the unique correspondence of points in the tip and hub sections. The tip section was then connected to the hub section by linear nodal coordinate generation. Connectivities were generated from the leading to the trailing edge. This scheme took maximum advantage of the automatic generation features of SAP IV and ADINA.

B. THE NEGATIVE JACOBIAN DETERMINANT

As mentioned earlier, the leading and trailing edge elements were ill-behaved. The leading and trailing edges of

this blade are arcs of a circle. When an element with quadratic sides is used, these elements are prone to re-entrant angles, and resultant negative or zero determinants of the Jacobian matrix. In other words, the inverse of the Jacobian matrix does not exist because there is not a unique correspondence between the natural and local coordinates of the element [8]. This will stop the program because the inverse of the Jacobian matrix is used in the strain-displacement matrix [B]. Recall:

$$\{\epsilon\} = [B]\{u\}$$

$$[B] = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} [H]$$

$$\begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{Bmatrix}$$

Also: $[K] = \int_{VOL} [B]^T [C] [B] d(VOL)$

$[J]^{-1}$ = inverse of the Jacobian matrix

$[K]$ = stiffness matrix

$\{u\}$ = $[H]\{u_i\}$ - displacement field

$[H]$ = matrix of shape functions

$[C]$ = matrix of elastic properties

$\{\epsilon\}$ = strain matrix

r,s,t = natural coordinates

As the above equations show, the inverse of the Jacobian is an essential part of the calculations for the strain matrix, and the stiffness matrix. Experience has also shown that when the determinant of the Jacobian is small (10^{-7} or smaller) the displacements at that node may lose almost all their numerical significance.

C. MESH VERIFICATION THROUGH PSAP1

Kibler [11] implemented a finite element preprocessor for mesh verification for the SAP IV and ADINA programs. This enables the user to check visually the coordinates and geometry of a mesh by generating graphical displays of the completed mesh. Oblique orthographic projections are produced using the NPS Calcomp model 765 plotter. Other displays are possible.

The PSAP1 program is especially useful because of the various mesh presentation options in the program. Among them are:

1. Node numbering
2. Element numbering

3. Exploded plots
4. Full rotation about each axis
5. Sectioned plots based on geometry,
node number, or element number
6. Displacement postprocessing using either
the deformed structure, or displacement
vectors at the nodes. (See Chapter VII)

Other options, as well as a User's Manual, are fully discussed in Ref. [11].

Figures 16 through 19 show the various mesh presentations, for the linear airfoil sections. PSAP1 was used extensively in the production of the airfoil mesh. The clear graphical presentation assures the engineer that he is using a mesh free of geometric or connectivity errors.

The elements with negative Jacobians were separately selected from the mesh and greatly magnified. The re-entrant angles were clearly visible on these plots as shown in Fig. 20. Large scale plots of the hub and tip cross-sections for these elements were made by hand. The mid-size nodes were moved inward toward the geometrical center of these elements.

Two airfoil meshes were developed. One was composed of 20-node bricks, and the other of 12-node bricks. The 12-node bricks were oriented such that the quadratic side could be used to model the airfoil camber. This orientation was the same as the one used by Garrett. Since the 12-node brick has only one quadratic side, part of the leading and trailing edge curves were eliminated. This is shown in Fig. 21. Aside

from this difference, the geometry represented by the two meshes was identical. The 12-node airfoil is shown in Fig. 22.

The characteristics of each of these meshes are shown in Table I.

V. AIRFOIL CASE STUDY

After thoroughly investigating the ceramic bar problem, the next step was to analyze linear airfoil section of the ceramic gas turbine blade using the ADINA code 12-node and 20-node bricks. The bar problem showed that significant differences occurred as a result of the brick type and orientation used. The purpose of the airfoil study was to determine the difference, if any, between the 12-node brick airfoil and a 20-node brick airfoil. Each airfoil was loaded using the actual design point conditions that could be expected in the ceramic gas turbine. Two general types of loading were investigated: centrifugal and thermal.

A. CENTRIFUGAL LOAD

A centrifugal load preprocessor was developed at NPS, and is discussed fully in Ref. [12]. The preprocessor can be used to generate consistent nodal loads for a given angular speed. The rotor design speed of 41,730 RPM was used.

Figures 23, 24 show the deformed foil superposed on the undeformed foil. (Chapter VII discusses using PSAP1 to plot the deformed structure.) The root plane coincides exactly in all cases because the nodes there were fully constrained. Two general deformations can be seen: first, there is a general elongation of the blade in the radial direction, and

second, there is an untwisting of the blade. Figures 25-26 show the deformed 12-node and 20-node foils superposed on each other. It can be seen that the 12-node model is stiffer in representing the untwisting deformation. Tables II and III show comparisons of stresses and maximum principal stresses, as calculated by the 12-node and 20-node bricks, at selected nodes in the root section. At each of the tabulated nodes, the stresses predicted by the 20-node brick are higher than those predicted by the 12-node brick. The stress contours for the root section are shown in Figs. 29 and 30.

B. THERMAL LOAD

Reference [13] shows the steady state isotherms. For this study, the isotherms were modeled as constant for various radial distances along the foil. The cited reference shows that this is a good assumption in the root section, where the isotherms are flat and closely spaced. In the upper portion of the foil, however, the isotherms have a definite curvature in the radial direction. Figure 31 shows the nodal temperatures used in this study. Preparation of the nodal temperature tape is discussed fully in Appendix A.

Figure 32 shows the deformed foil superimposed on the original geometry. A general enlargement of the foil in all directions is seen. Table IV shows a comparison of thermal stresses at selected points. (It should be noted that these are not evaluated at nodal points, but rather at the Gauss points. ADINA does not give nodal stresses for thermal loadings.)

C. CONCLUSIONS FROM AIRFOIL STUDY

This comparative study shows how the choice of a finite element can affect the results. The 12-node brick, even though oriented in the most advantageous way, is stiffer than the fully quadratic three-dimensional element. The stiffer element produces lower stresses in the root section. It is these root section stresses which transmit the tensile forces and moments to the blade platform and eventually to the critical dovetail area.

VI. DISTRIBUTED LOAD PREPROCESSOR

The version of ADINA installed at NPS does not contain a distributed load feature. Finite element texts, such as Ref. [14], often state a consistent allocation of a uniform surface load for a rectangular element. The consistent load for a curved surface, such as the airfoil studied in this report, is more complex. Since a distributed load is used to model the aerodynamic pressure on an airfoil, a program to calculate consistent pressure loads for curved surfaces was necessary.

A. DEVELOPMENT OF THEORY

A consistent load is defined as a set of nodal forces which produce the same amount of work as the original force. Stated mathematically:

$$\langle v_i \rangle \{u_i\} \equiv \int_{\text{AREA}} \langle f^S \rangle \{u\} d(\text{AREA})$$

where: $\langle v_i \rangle$ = vector of nodal consistent loads

$\{u_i\}$ = nodal displacements

$\langle f^S \rangle$ = vector of surface forces

$\{u\} = [N_i] \{u_i\}$ = displacement field

$[N_i]$ = shape functions

Consider the general surface shown in Fig. 33.

$$\text{Work} = \int_A \underline{n} \, p dA \cdot \underline{u} = \langle v_i \rangle \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \quad (1)$$

\underline{n} = unit normal to the surface (positive outward).

p = pressure on dA .

\underline{u} = displacement vector = $u_i \underline{i} + v_j \underline{j} + w_k \underline{k}$

For ease of computation, it is desirable to perform the integration in the natural coordinate system. The vector $\underline{n}dA$ is transformed to the natural coordinate system by:

$$\underline{n}dA = \left(\frac{\partial \underline{R}}{\partial \alpha} \times \frac{\partial \underline{R}}{\partial \beta} \right) d\alpha d\beta$$

where: $\underline{R} = x_i \underline{i} + y_j \underline{j} + z_k \underline{k}$ = position vector of point on surface as shown in Fig. 33.

$d\alpha d\beta$ = differential area in natural coordinate system.

α, β are dummy variables standing for the appropriate natural coordinates which define the pressure face (See Table V).

$\frac{\partial \underline{R}}{\partial \alpha}$ This is a vector tangent to a line α where β is constant. The components of this vector are elements of the Jacobian matrix.

$\frac{\partial \underline{R}}{\partial \beta}$ is similarly defined.

The pressure and displacement fields can be discretized with the same shape functions:

$$p = \langle N_i \rangle \{p_i\} \quad (2)$$

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \quad (3)$$

where: N_i = shape functions
 $\{p_i\}$ = matrix of nodal pressures
 $\begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}$ = matrix of nodal displacements

Let:

$$\left(\frac{\partial R}{\partial \alpha} \times \frac{\partial R}{\partial \beta} \right) d\alpha d\beta = (\lambda_1 \underline{i} + \lambda_2 \underline{j} + \lambda_3 \underline{k}) d\alpha d\beta \quad (4)$$

then:

$$\underline{n} dA \cdot \underline{u} = \langle \lambda_1 \lambda_2 \lambda_3 \rangle \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} d\alpha d\beta$$

$$\begin{aligned} \lambda_1 &= x_{2,\alpha} x_{3,\beta} - x_{3,\alpha} x_{2,\beta} \\ \lambda_2 &= x_{3,\alpha} x_{1,\beta} - x_{1,\alpha} x_{3,\beta} \\ \lambda_3 &= x_{1,\alpha} x_{2,\beta} - x_{2,\alpha} x_{1,\beta} \end{aligned}$$

Using Eqs (2 through 5), Eq (1) can now be written

$$\langle v_i \rangle \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} = \int_{-1}^1 \int_{-1}^1 \langle \lambda_1 \lambda_2 \lambda_3 \rangle \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \langle N_i \rangle \{p_i\} d\alpha d\beta$$

$$\langle v_i \rangle = \int_{-1}^1 \int_{-1}^1 \langle \lambda_1 \lambda_2 \lambda_3 \rangle \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} \langle N_i \rangle \{p_i\} d\alpha d\beta \quad (6)$$

B. PROGRAM STRUCTURE

The consistent pressure load program is structured as follows:

Subroutine TRANS

- reads element number, coordinates, and connectivity
- reads the face to which pressure applied (NFACE)
- reads pressure at each node of NFACE
- reads number of Gauss points to be used in surface integration (2-6)
- calls CUBAT

Subroutine CUBAT

- establishes vector of weights and sampling points to be used in Gaussian integration
- calls SHAPE

Subroutine SHAPE

- evaluates shape functions at Gauss points
- evaluates derivative of shape function at Gauss point
- forms Jacobian matrix
- evaluates determinant of Jacobian matrix to check for zero or negative Jacobian determinant
- calls FACEP

Subroutine FACEP

- establishes matrices in integrand
- performs matrix multiplication to establish integrand
- performs surface integration

A listing of the consistent pressure load program is in Appendix C.

C. PROGRAM VERIFICATION

Several test cases were run to insure the accuracy of the pressure load preprocessor. A uniform distributed surface load was rotated through all faces of a one inch cube, and the 2-6 Gauss point integration scheme was checked for each face. The results agreed exactly, both in magnitude and sign, with the consistent load allocation found in Ref. [14].

The program was successfully checked to insure that distributed surface force could be applied to more than one face of the element simultaneously, and that pressures could be applied simultaneously to faces that share a common edge.

The ceramic flexure specimen mesh (Case 4) was also used to verify an actual problem. Pressure loads were applied to various faces of the specimen, including adjoining faces. In each case, the program produced correct results.

As a final test, a distributed load was also placed across the span between the supports. A sketch of the

problem is shown in Fig. 34. The results were again compared with simple beam theory. The results were excellent as shown in Fig. 35.

D. USER'S MANUAL

Figure 36 shows a hexahedral element in the natural coordinate system. The node numbering convention is also shown in Fig. 36. Table V shows the face numbering convention used in the pressure load preprocessor. Each face has four corner nodes and four midside nodes. The pressure distribution acting on an element face is defined by specifying the nodal pressure intensities at the corner and mid-side nodes. For example, if Face 4 were loaded P_4 , P_8 , P_7 , P_3 , P_{20} , P_{15} , P_{19} , and P_{11} must be specified in that order. Appendix B lists the input data and results for the sample bar problem, shown in Fig. 36.

1. Control Card (10I5)

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
1-5	NUMNP	Total number of node points in problem.
6-10	NUMEL	Total number of elements in problem.
11-15	NGP	Number of Gauss points to use in surface integration. ($2 \leq \text{NGP} \leq 6$)
16-20	NCUR	Number of ADINA load curve that output loads refer to. (usually, NCUR=1)

21-25	NPLOAD	Number of element faces with distributed loads.
26-30	IPRINT 1	1-program will just read and print ADINA input deck
31-35	IPRINT 2	1-program will not print element-by-element solution.
36-40	IPRINT 3	1-program will not punch output deck of consistent loads.
41-45	IPRINT 4	1-program will omit printing input data.

2. Nodal Pressures Card (215, 8F5.0)

1-5	NPREL (J)	element number
6-10	NFACE (J)	face to which pressure applied. (See Table V)
11-50	PRESS (I,J)	nodal pressure intensities input here: corner nodes first, then midside. (See Table V) Positive pressure is directed in same sense as outward unit normal.

E. USE OF PRESSURE LOAD PREPROCESSOR ON CP/CMS

The pressure load preprocessor is ideally suited for use on CP/CMS. A compiled version of the program (filetype=text) should be stored on a private disk or workspace. A data file, such as FT04F002, should also be defined. The input data deck (Appendix B) can then be read into this file [15].

It is suggested that an executive type routine also be used:

```
& TYPEOUT ERROR
FILEDEF 05 DSK FILE FT04F002 (PERM)
FILEDEF 06 PRT (PERM)
FILEDEF 07 PUN (PERM)
VSET RDYMSG OFF
$ PRESSURE
```

This defines your input/output files and executes the program. Output is directed to the line printer, and card punch.

The pressure program utilizes approximately 300K BYTES of core. The extra core should be requested at LOGIN by:

```
L^nnnnPtt^300K
```

where:

```
nnnn = user number
tt = terminal number
```

VII. USING PSAP1 AS A DISPLACEMENT POSTPROCESSOR

PSAP1 is capable of postprocessing displacements. The results can be presented in two ways:

1. The deformed structure can be plotted. In this case, scaled displacements are added to the original geometry to produce the deformed coordinates. The deformed structure is then plotted using this new geometry.
2. The displacements can be represented as scaled vectors. The scaled displacement vectors are plotted at original geometry nodes.

The purpose of this chapter is to explain the modification of the ADINA code to store the displacements, and how to use the PSAP1 postprocessing capabilities.

A. MODIFICATION OF ADINA

Reference [16] states that a version of SAP IV had been modified to punch a deck of displacement cards from the output data. When dealing with large quantities of data, however, storage of this data on direct access storage devices is preferable. Therefore, the ADINA code was modified to store the output displacements on device 58, in addition to the usual printed output.

The flag for saving the displacements is the integer number 1 in column 80 of the ADINA master control card [4]. The flag variable name is ITP 58, and is passed in common /EAST/.

The job control language (JCL) for ADINA must also be modified to define device 58:

```
//FT58F001 DD DISP=OLD,UNIT=3330,VOL=SER=DISKO,  
//          DSN=Snnnn.pppp
```

where:

nnnn is the user number

pppp is the name assigned by the user to identify the data set.

Since an unformatted write statement is used to store the data on device 58, a variable blocksize must be used when allocating disk space. Reference [17] explains this procedure.

B. MODIFICATIONS OF PSAP1

There are presently two subroutines in PSAP1 that can be used to input displacement data to PSAP1.

1. Subroutine DATA 9 was added to PSAP1 by Losh [16]. This subroutine reads displacement data from punched cards. Reference [16] discusses its use, and should be consulted.

2. Subroutine DATA 5 reads displacement data directly from device 58. A JCL card must be added to the PSAP1 JCL to define device 58:

```
//GO.FT58001 DD DISP=OLD,UNIT=3330,VOL=SER=DISKO,  
//          DSN=Snnnn.pppp
```

C. USER'S MANUAL FOR DISPLACEMENT POSTPROCESSING

The PSAP1 User's Manual [11] applies with the following modifications:

a) Namelist Option

Variable, Value	Description
NUDISP = 1	x direction displacements input
NVDISP = 1	y direction displacements input
NWDISP = 1	z direction displacements input
KDATA = 5	subroutine DATA 5 used to read displacement data from device 58.
NVALUS = nnn	nnn is the integer number of displacements set to be plotted. This is equal to the number of node points.

b) Namelist Pict

KDISP = n	n=1: plot of deformed structure n=3: displacements represented by vectors at nodes.
IDMAG = n	n=1: direct magnification of displacement data by DMAGS. n=2: scaling of displacement data to a maximum value of DMAGS.
DMAGS = r	r is a floating point number which indicates the magnification of the displacements. (0.1 has been found appropriate.)

Figures 37 and 38 are examples of displacement postprocessing. The structure is the bending specimen loaded in four-point flexure.

Figure 37 illustrates the deformed structure option of PSAP1. Figure 38 shows the displacements of element 15 plotted as scaled vectors at the node points.

VIII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

1. The 12-node brick is a stiffer model of a structure than the fully quadratic 20-node brick.

2. The accuracy of the results obtained using a 12-node brick depends on geometry, loading, and the orientation of the element.

3. The flexure problem showed that two orientations of the 12-node brick are stiffer than the third. Resulting displacements and stresses differed significantly, and in all cases were lower than the 20-node brick and theory.

4. The airfoil study showed that the maximum principal stresses predicted in the root section could differ by as much as 14.9% depending on the basic element used. The 12-node brick under-predicted the 20-node results. The use of the 12-node brick could therefore produce an unsafe design.

B. RECOMMENDATIONS FOR FUTURE WORK

As is true with many projects, this thesis raised many new questions. The following areas of future work are suggested:

1. The ADINA code should be reviewed with an eye toward streamlining it. A basic code using only two- and three-dimensional elements, and the linear isotropic material model would be a useful tool. A reduced version of ADINA

could then be installed on the time-sharing system, or other stand-alone system.

2. The Field program, developed at NPS, should be examined and expanded if necessary. This program would serve as an excellent thermal preprocessor. It should be modified to produce a nodal temperature tape that could be used in ADINA, or other code capable of using temperature inputs.

3. Aerodynamic pressure loads should be examined. A program that could calculate the nodal pressure intensities for an element would provide the necessary input information for the distributed load preprocessor.

4. Centrifugal, pressure and thermal loadings should be superposed.

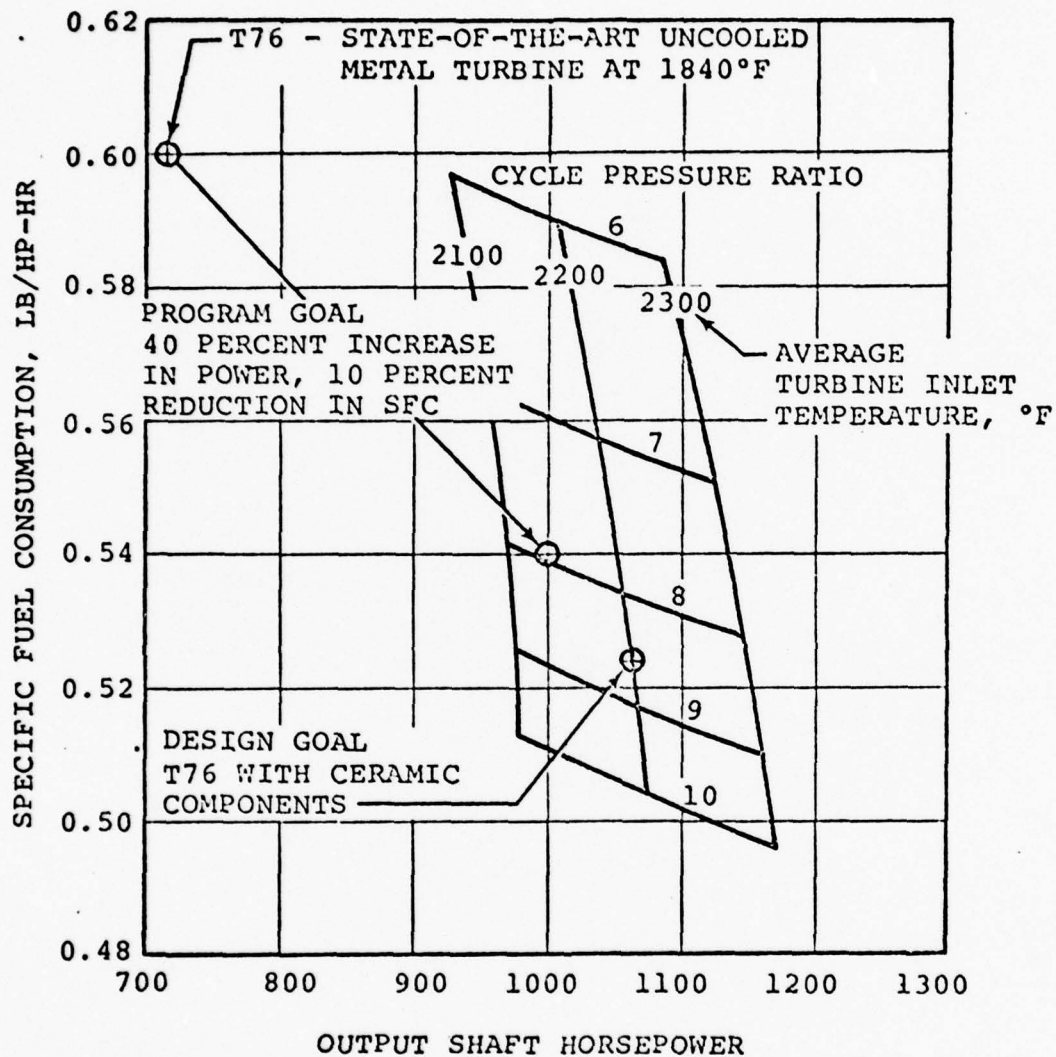


Figure 1. Engine Design Parameters

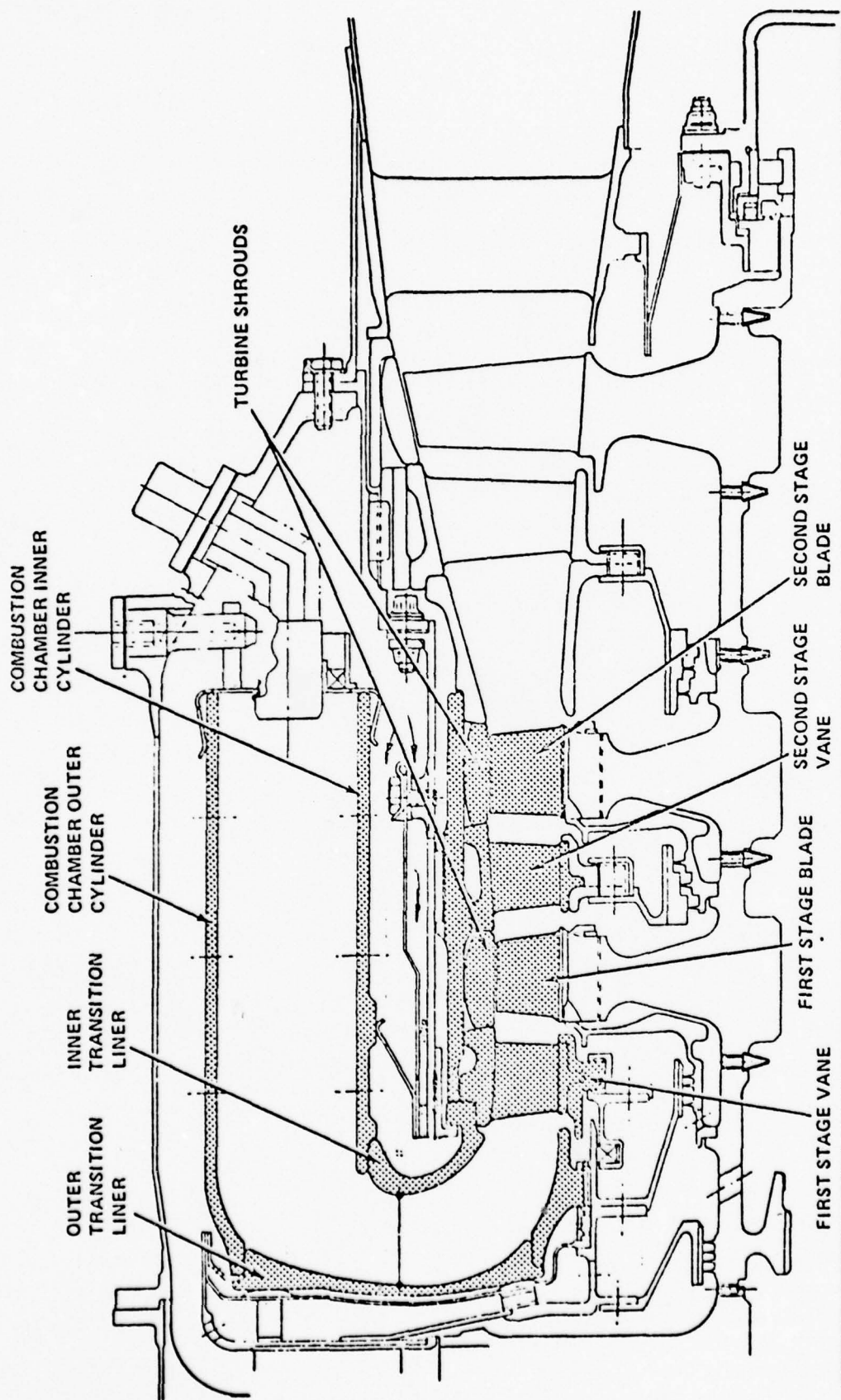
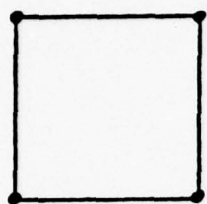
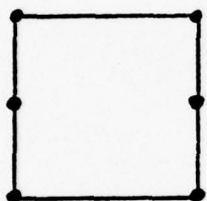
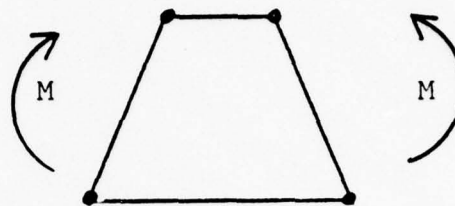


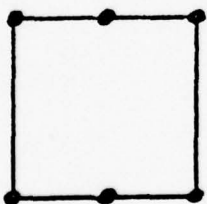
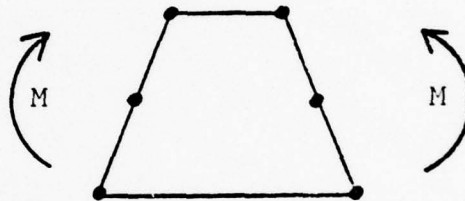
Figure 2. Ceramic Hot-Section Components



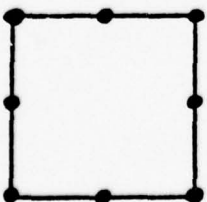
a. 4 Node Plane Element



b. 6 Node Plane Element: Orientation 1.



c. 6 Node Plane Element: Orientation 2.



d. 8 Node Plane Element

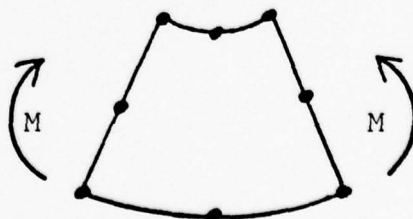
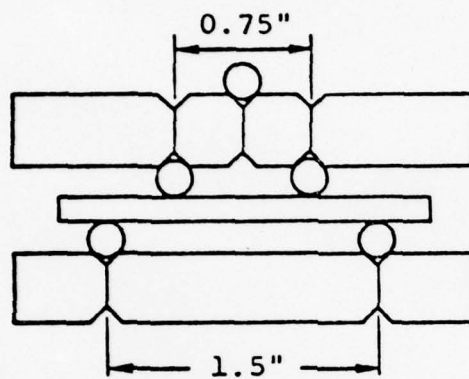


Figure 3. Two-Dimensional Plane Element



SCHEMATIC SHOWING SELF-ALIGNING CAPABILITY

Figure 4. Schematic of 4-Point
Flexure Rig

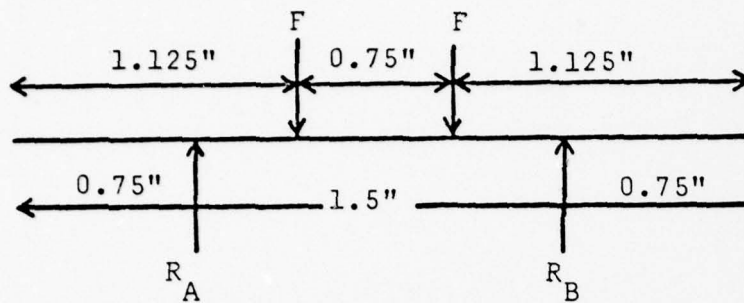


Figure 5a. Bend Specimen Free-Body Diagram

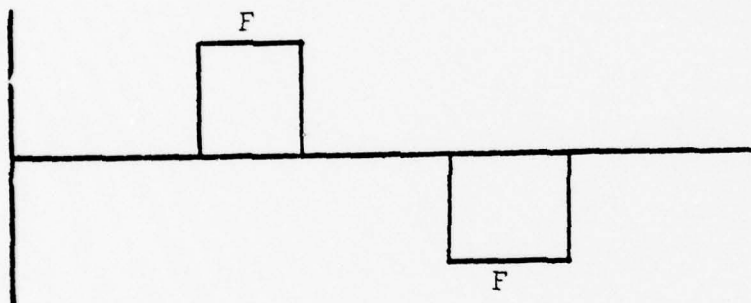


Figure 5b. Bend Specimen Shear Diagram

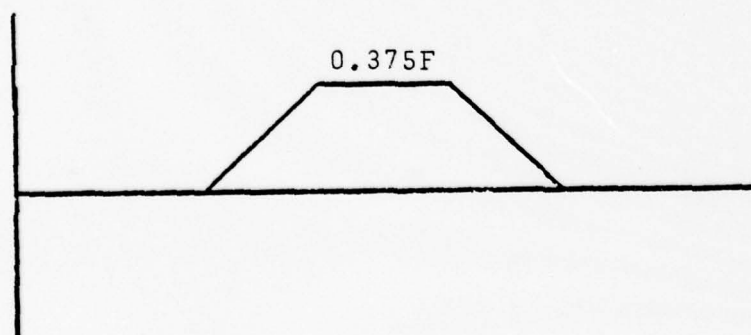


Figure 5c. Bend Specimen Moment Diagram

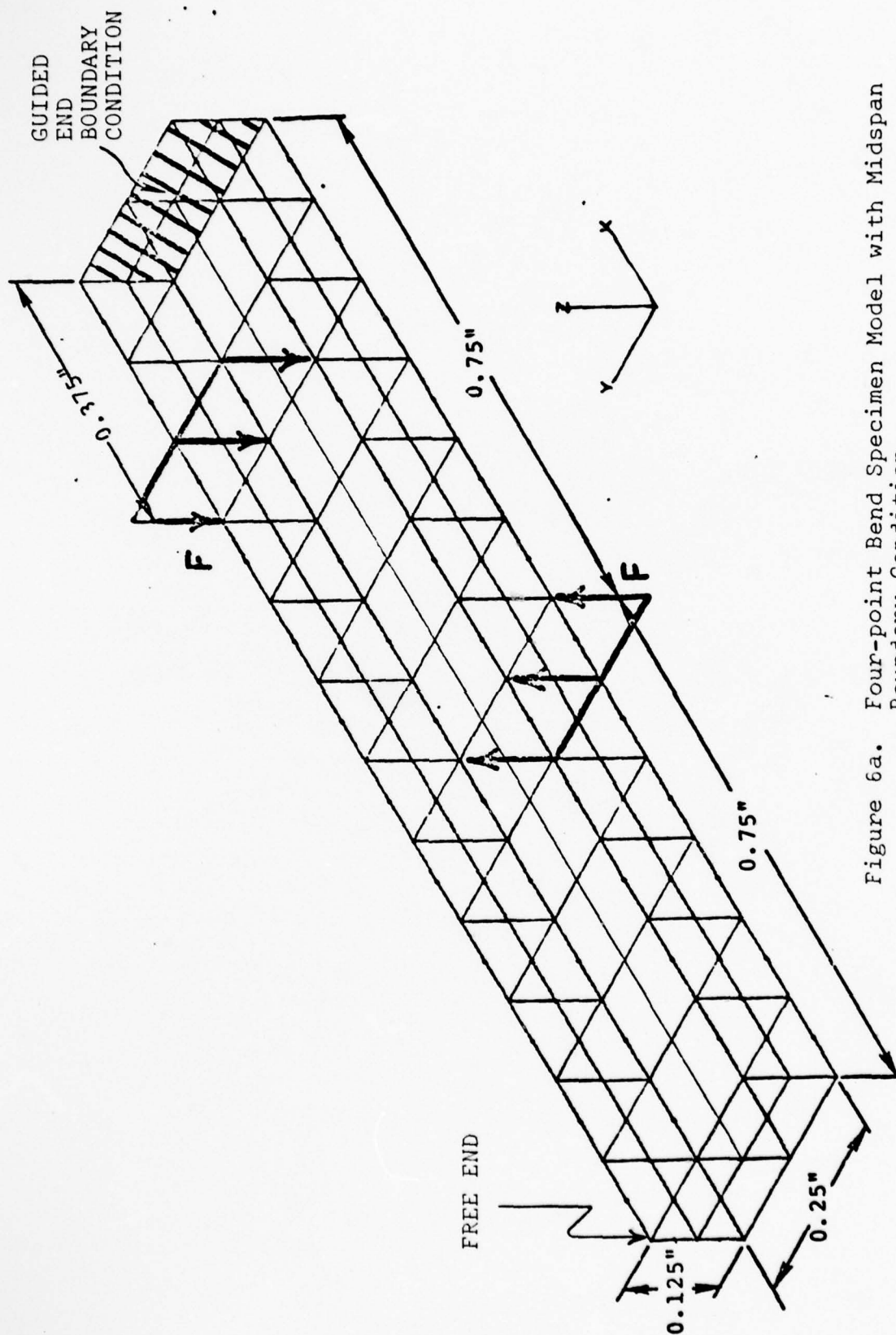


Figure 6a. Four-point Bend Specimen Model with Midspan Boundary Condition.

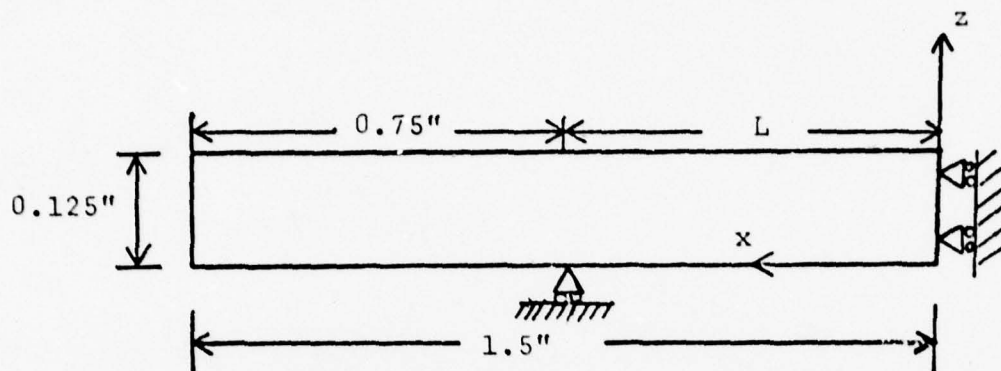


Figure 6b. x-z View of Half-Span with Guided End Condition

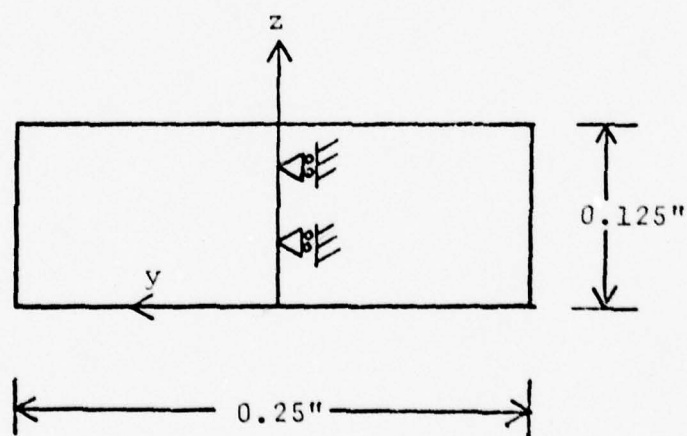


Figure 6c. y-z View of Specimen with Guided Midplane Boundary Condition

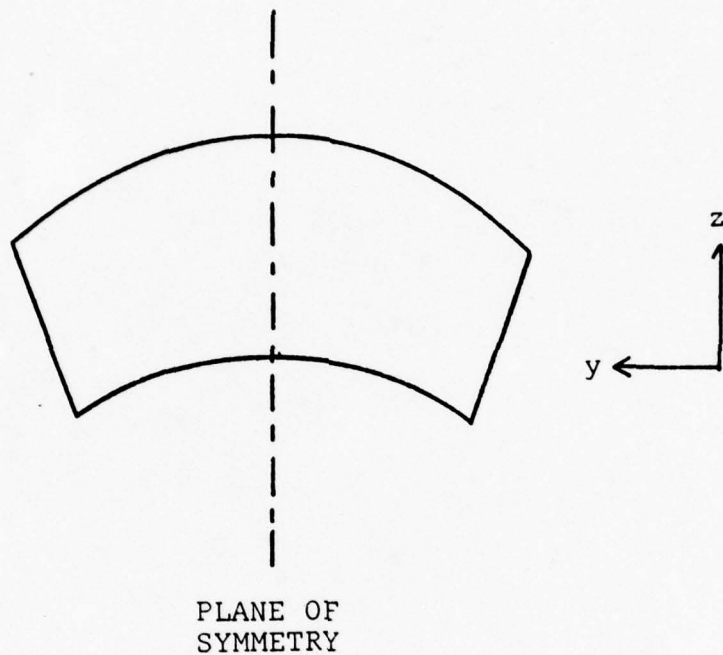


Figure 7. Cross-Sectional View of Deformed Specimen Showing x-z Plane of Symmetry

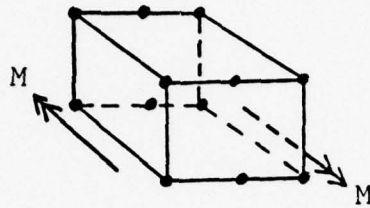


Figure 8a. 12-Node Brick Orientation for Case-1

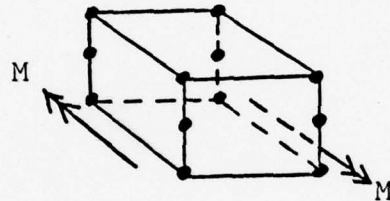


Figure 8b. 12-Node Brick Orientation for Case-2

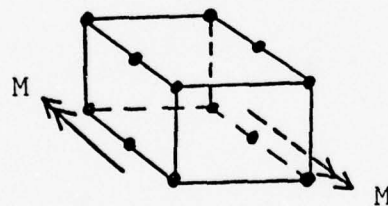


Figure 8c. 12-Node Brick Orientation for Case-3

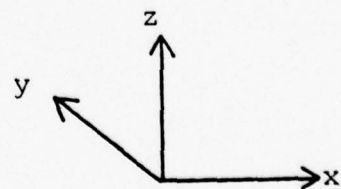
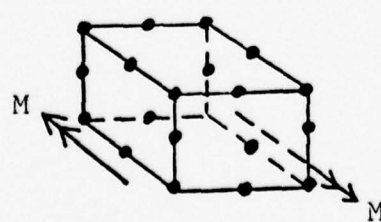


Figure 8d. 20-Node Brick; Case-4

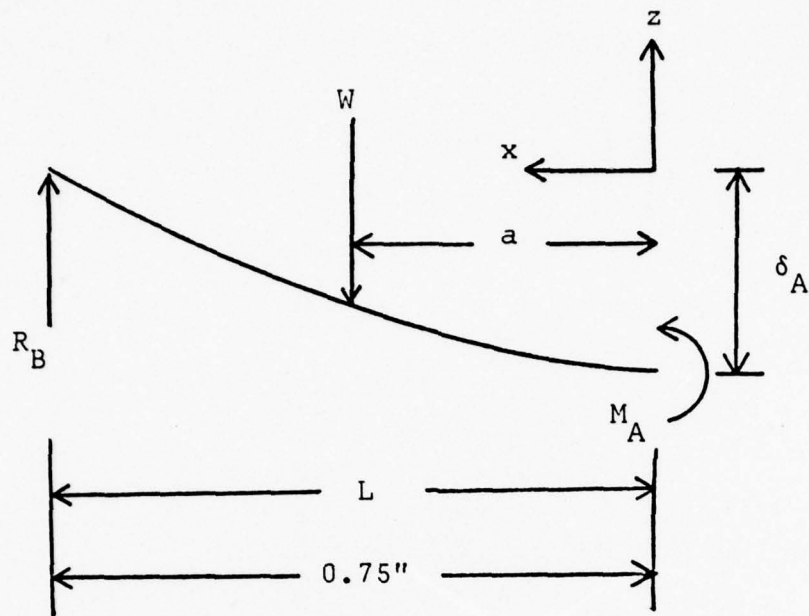


Figure 9a. Free Body Diagram for Flexure Case Study

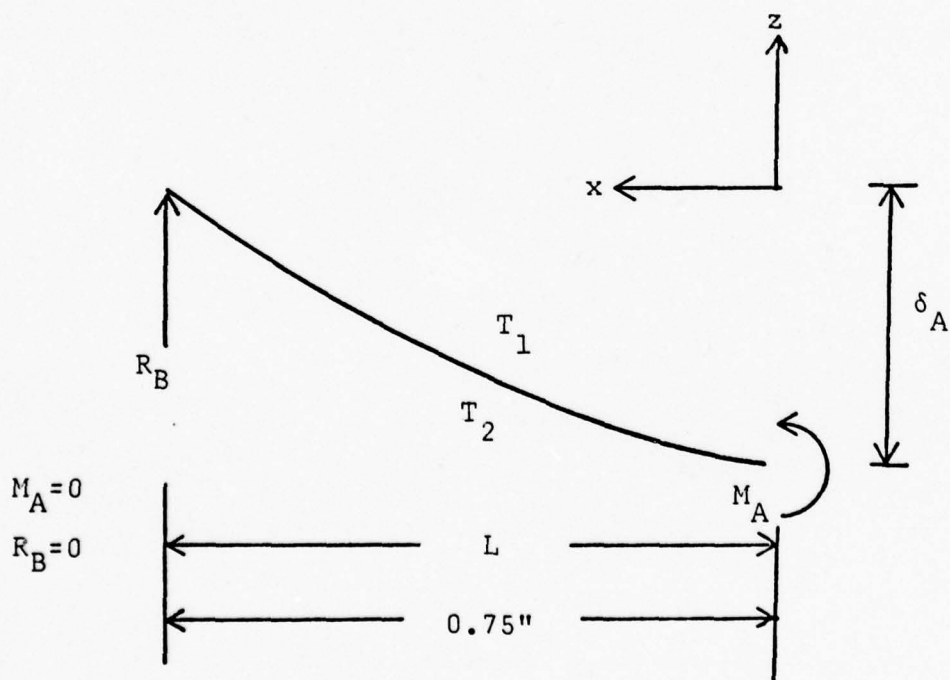


Figure 9b. Free Body Diagram for Temperature Gradient Case Study

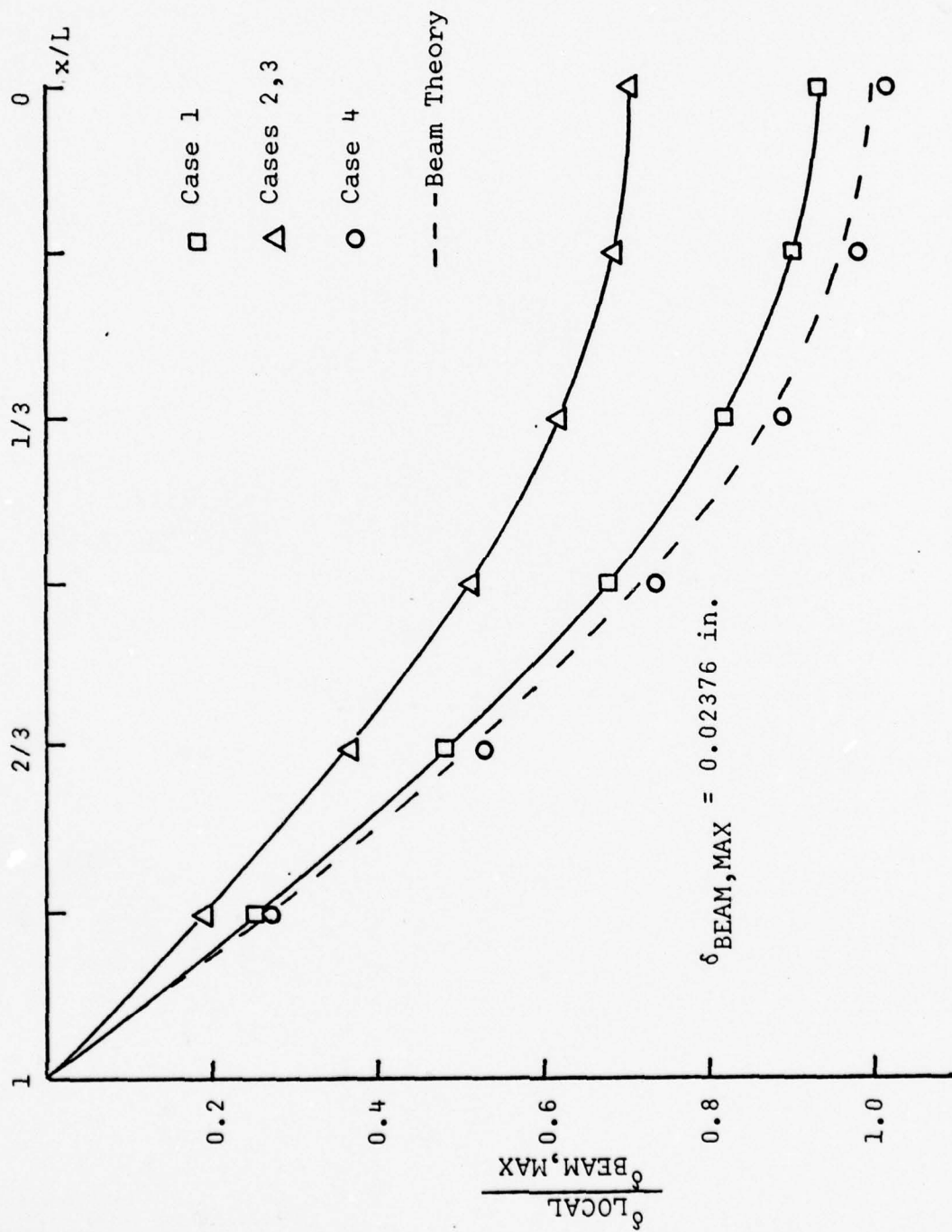


Figure 10. Normalized Displacement vs. Span for Steel Bend Specimen

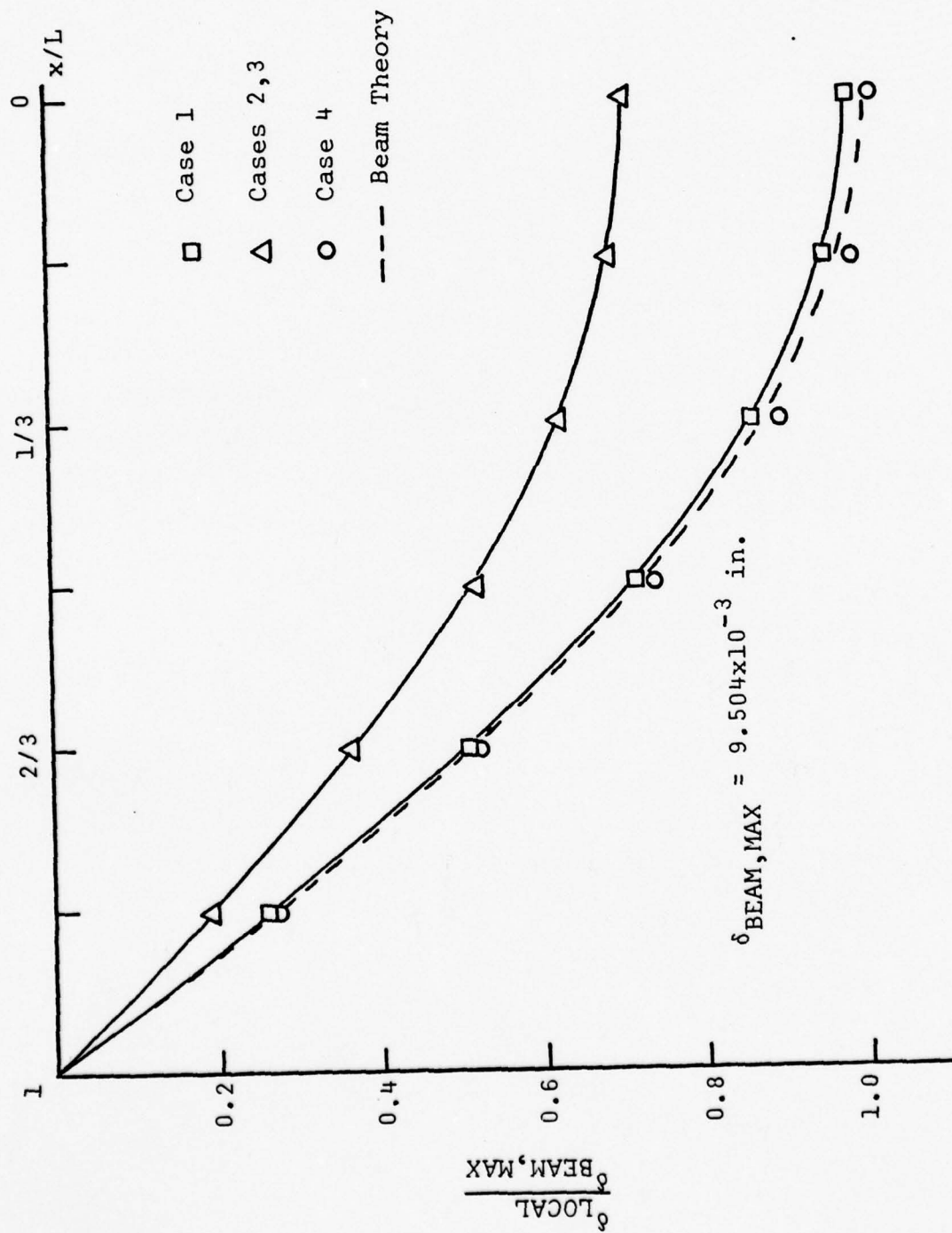


Figure 11. Normalized Displacement vs. Span for Ceramic Bend Specimen

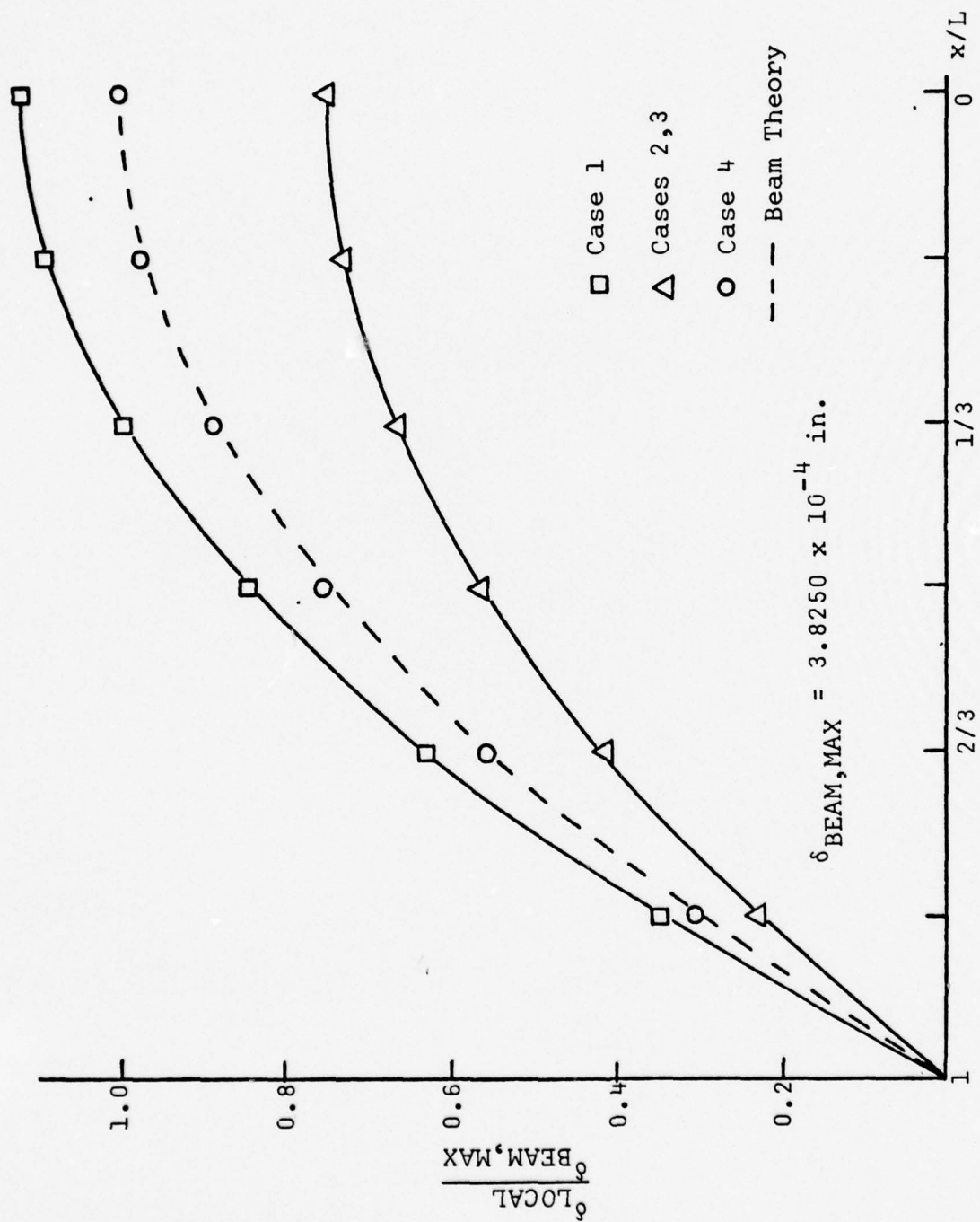


Figure 12. Normalized Displacement vs. Span for Temperature Gradient Analyses ($\nu=0.2$)

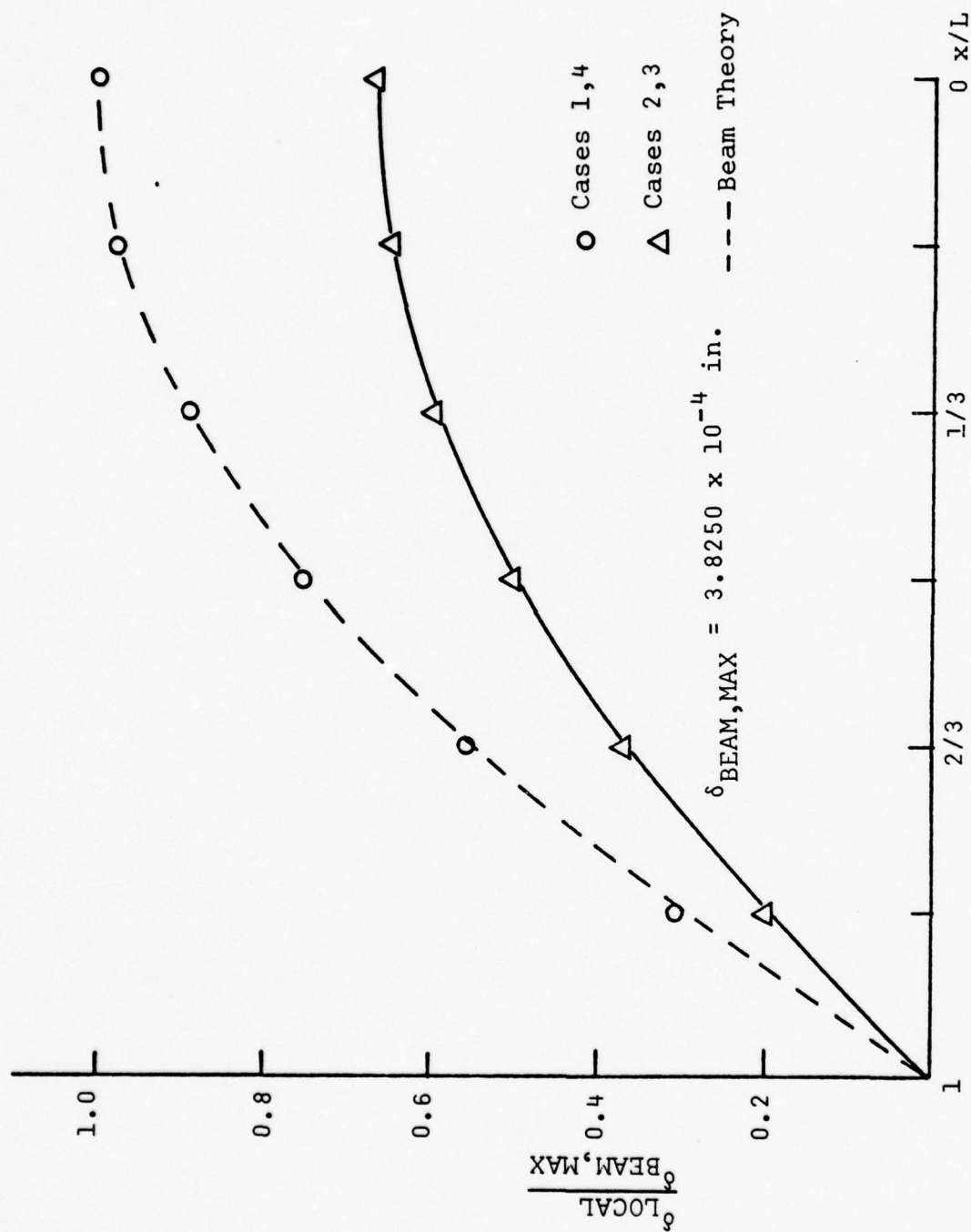


Figure 13. Normalized Displacement vs. Span for Temperature Gradient Analyses ($\nu=0$)

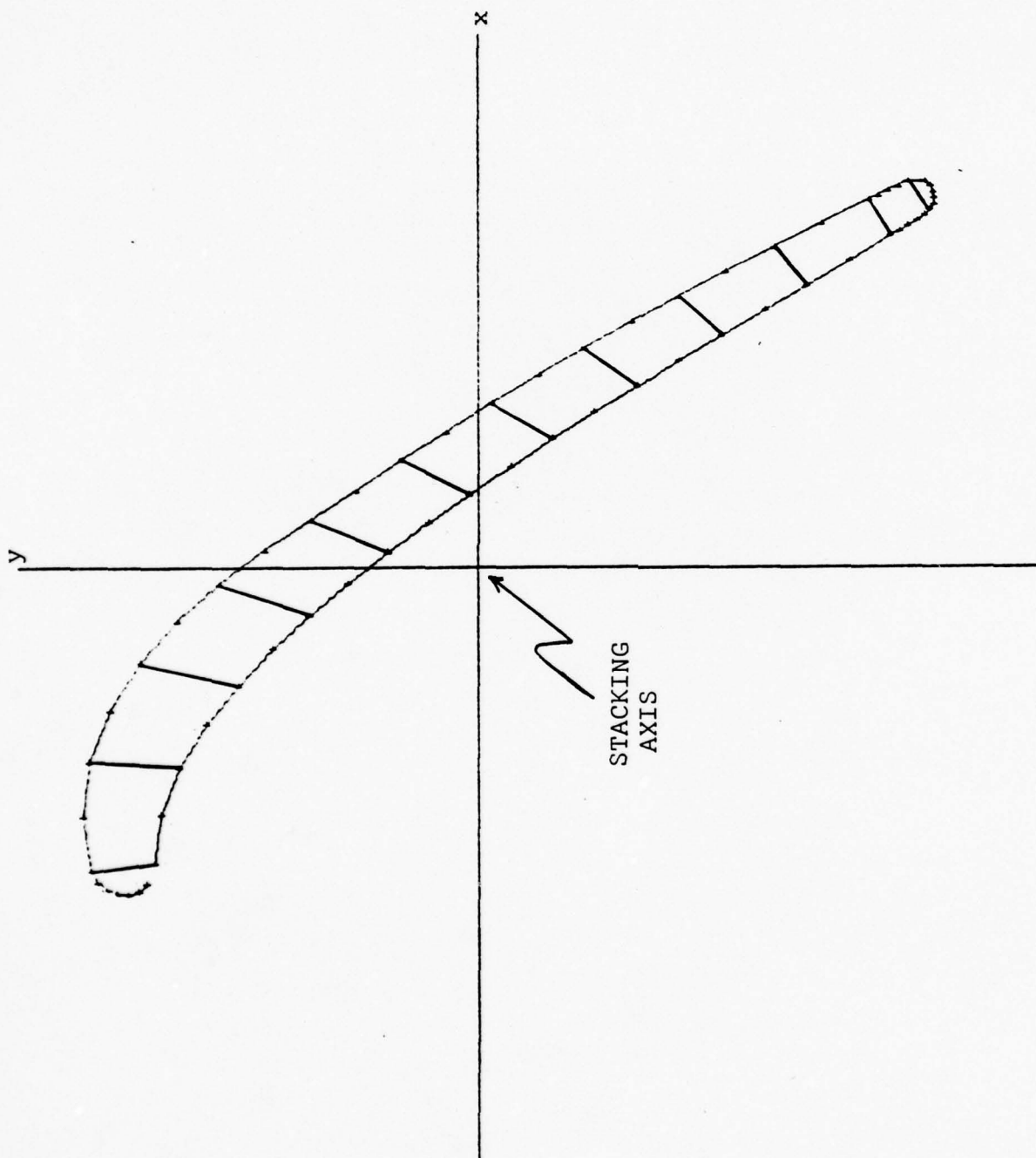


Figure 14. Airfoil Tip Section Profile

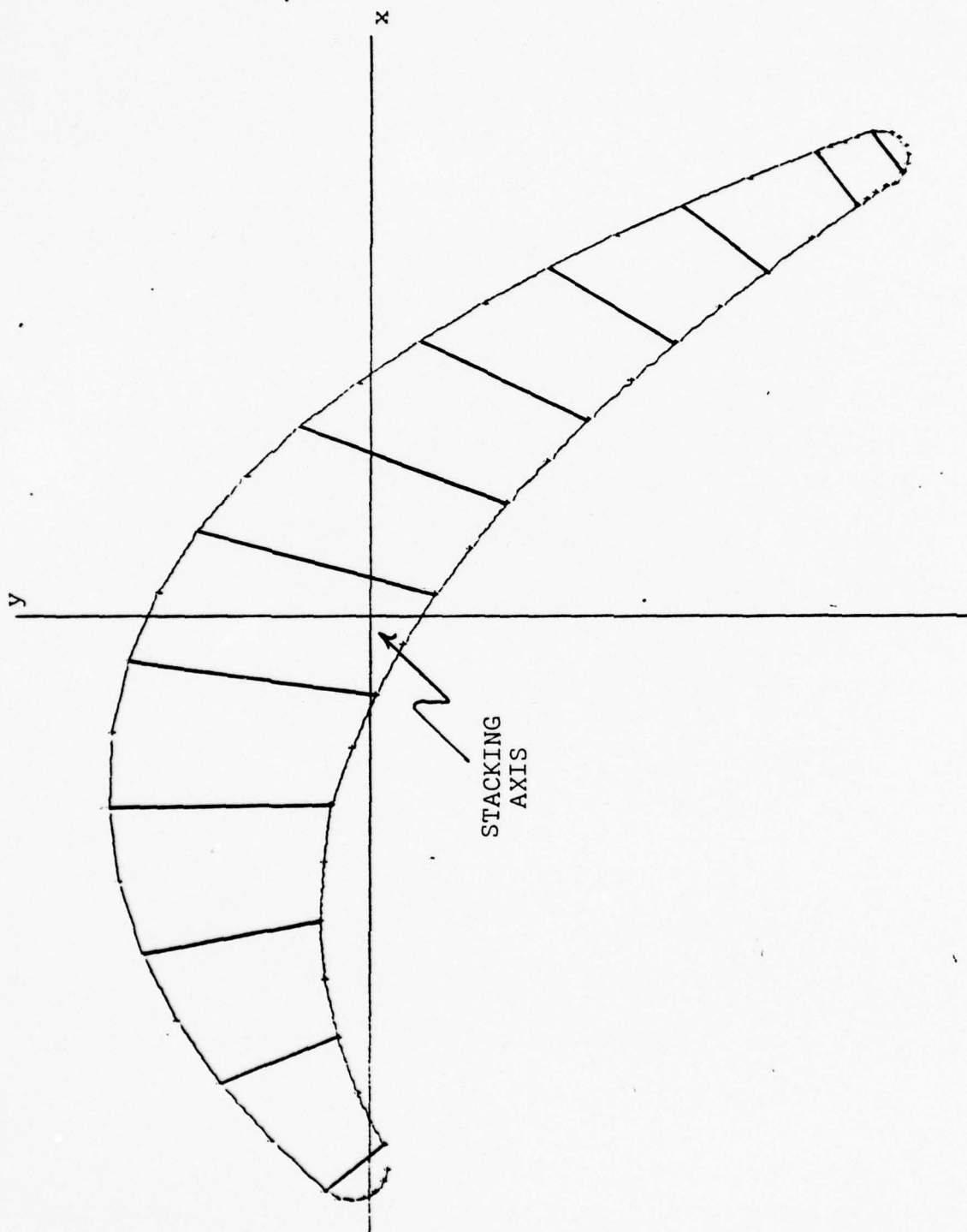


Figure 15. Airfoil Hub Section Profile

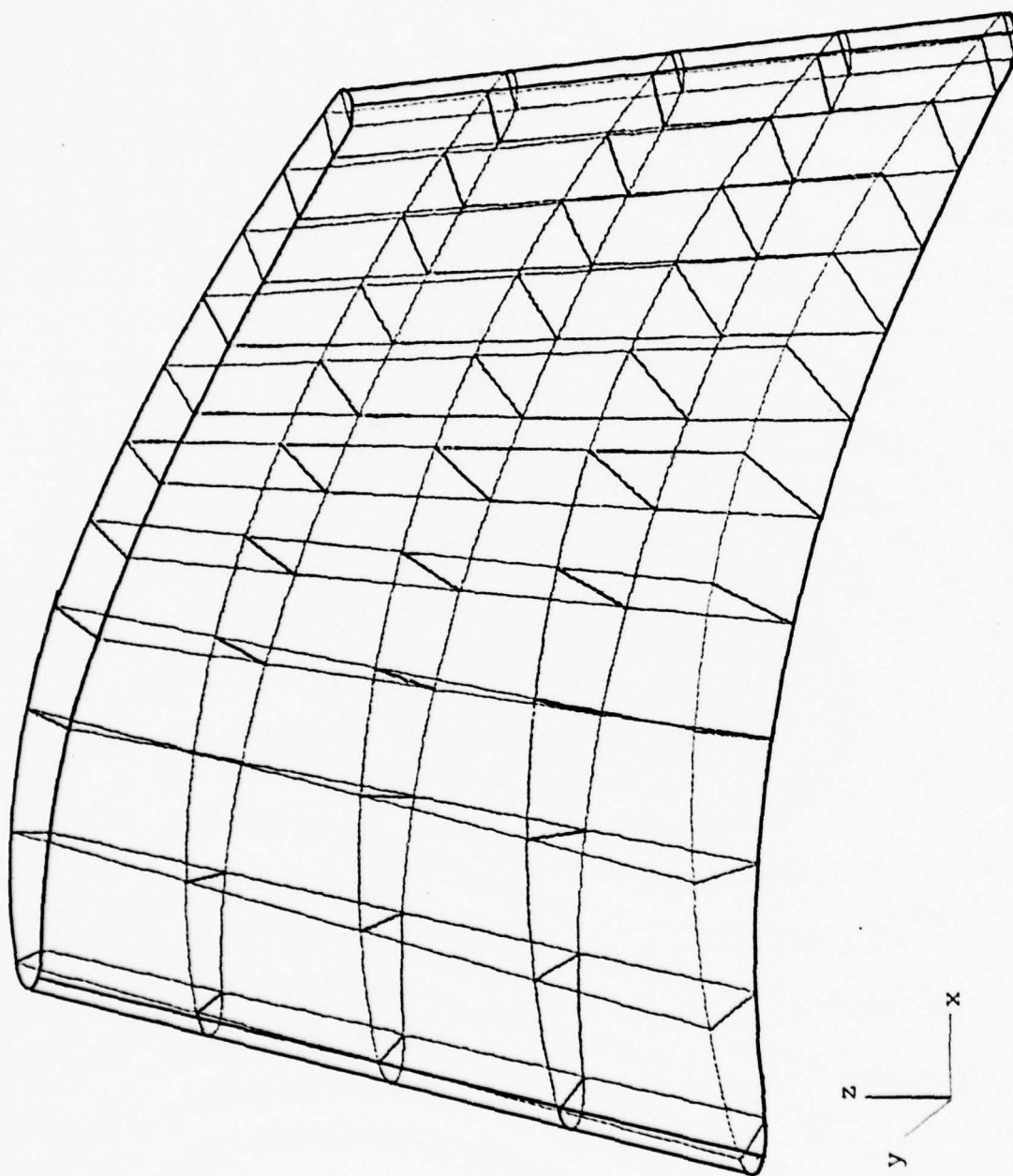


Figure 16. 20-Node Airfoil Mesh View 1

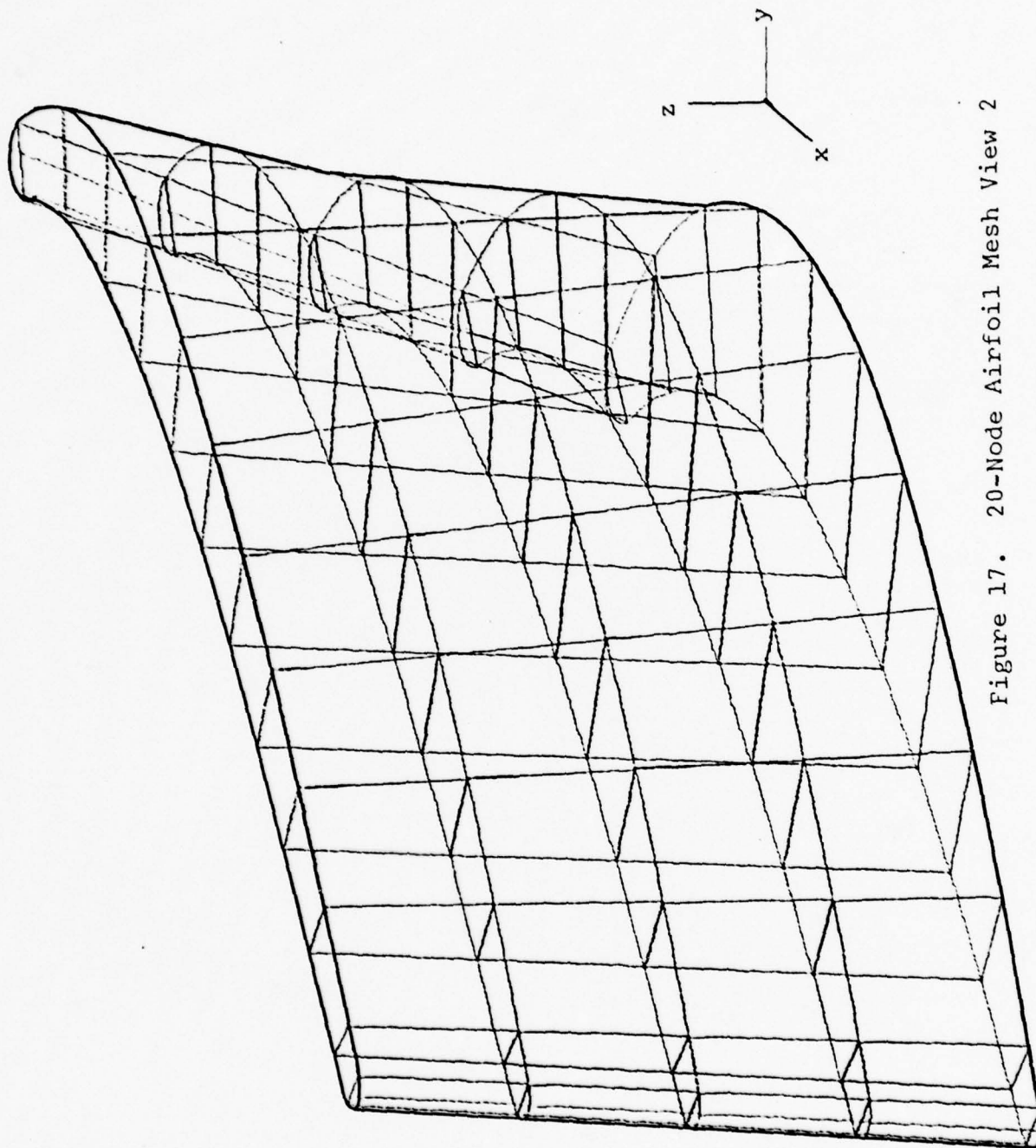


Figure 17. 20-Node Airfoil Mesh View 2

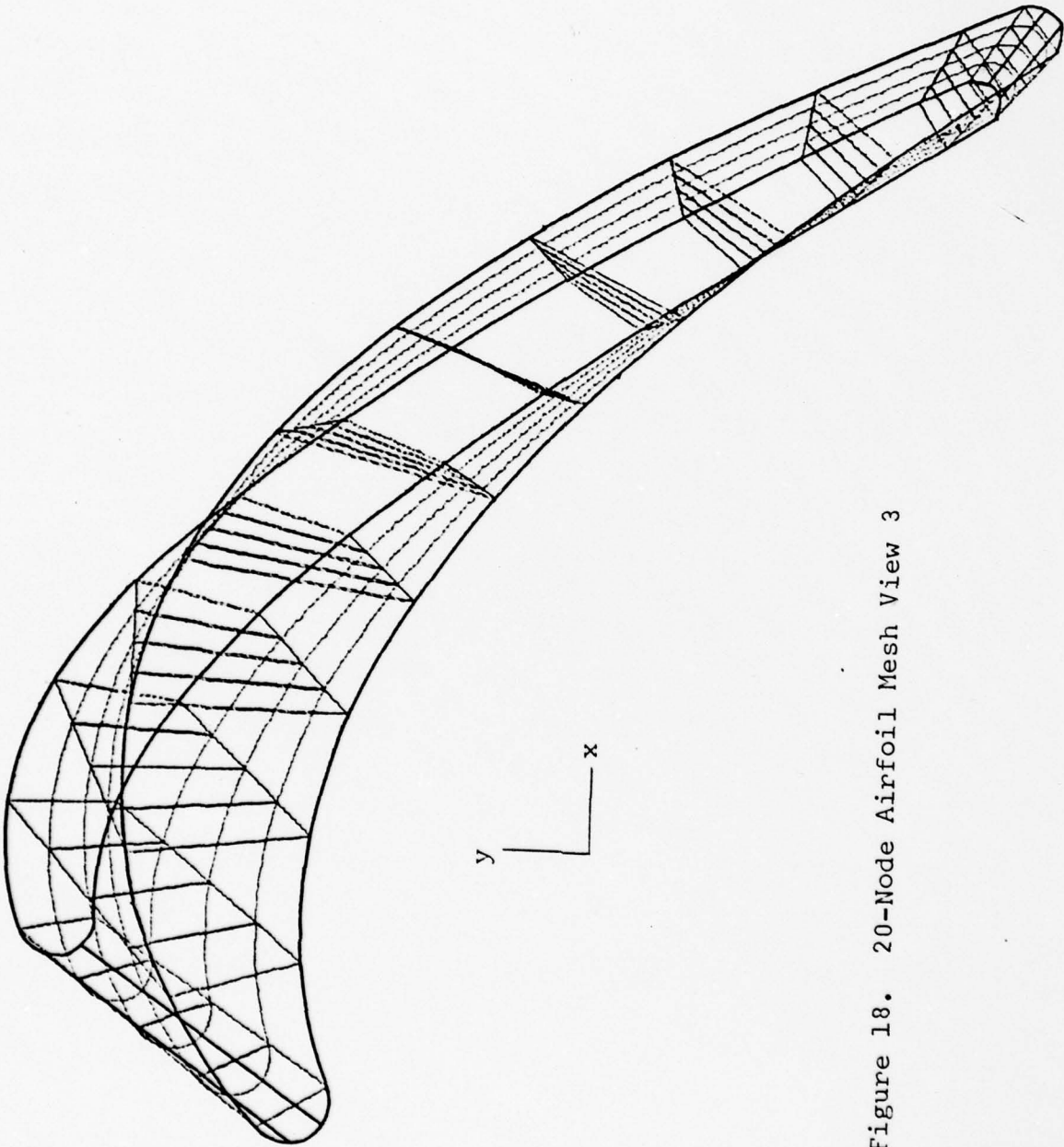


Figure 18. 20-Node Airfoil Mesh View 3

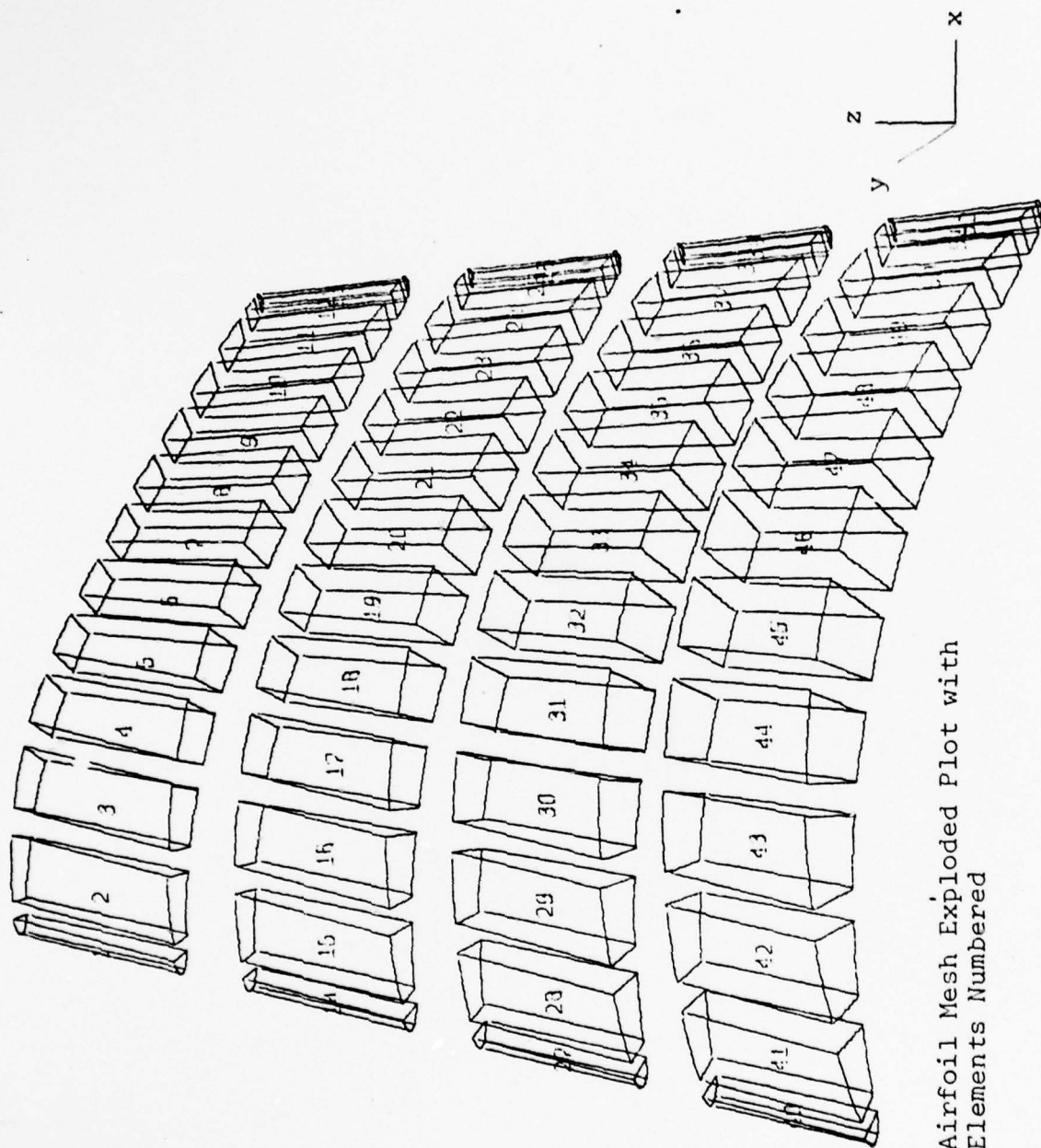


Figure 19. Airfoil Mesh Exploded Plot with Elements Numbered

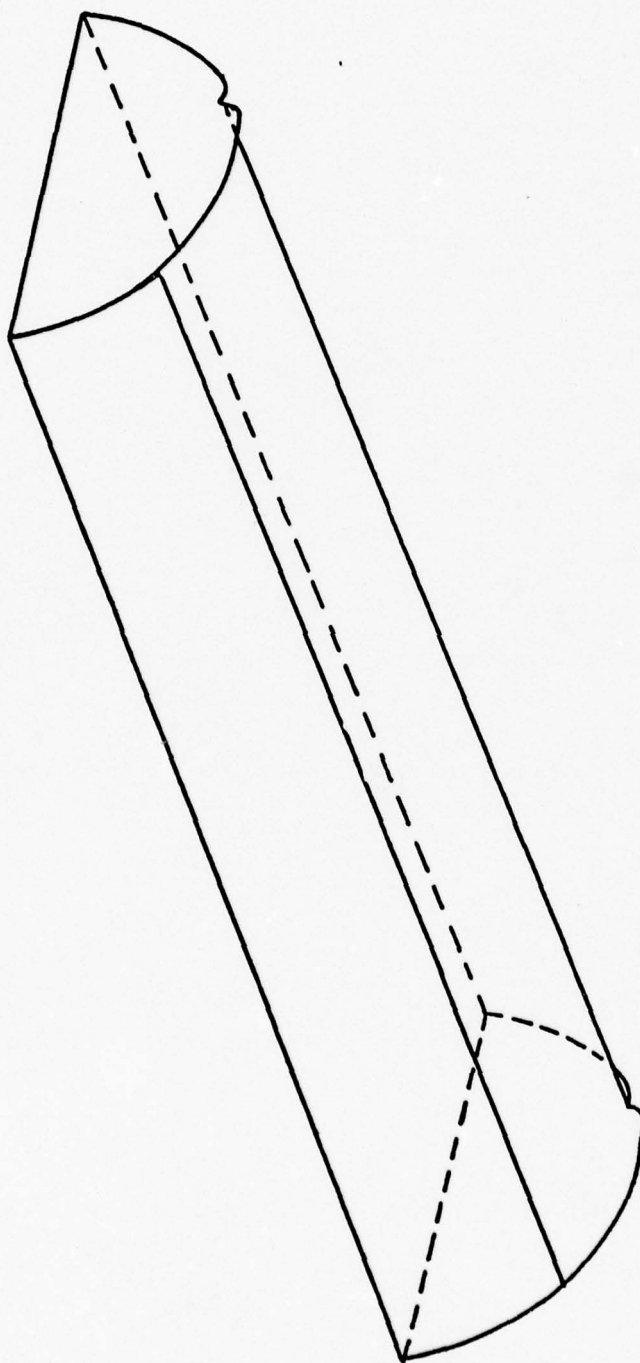


Figure 20. Trailing Edge Element Showing Re-entrant Angle

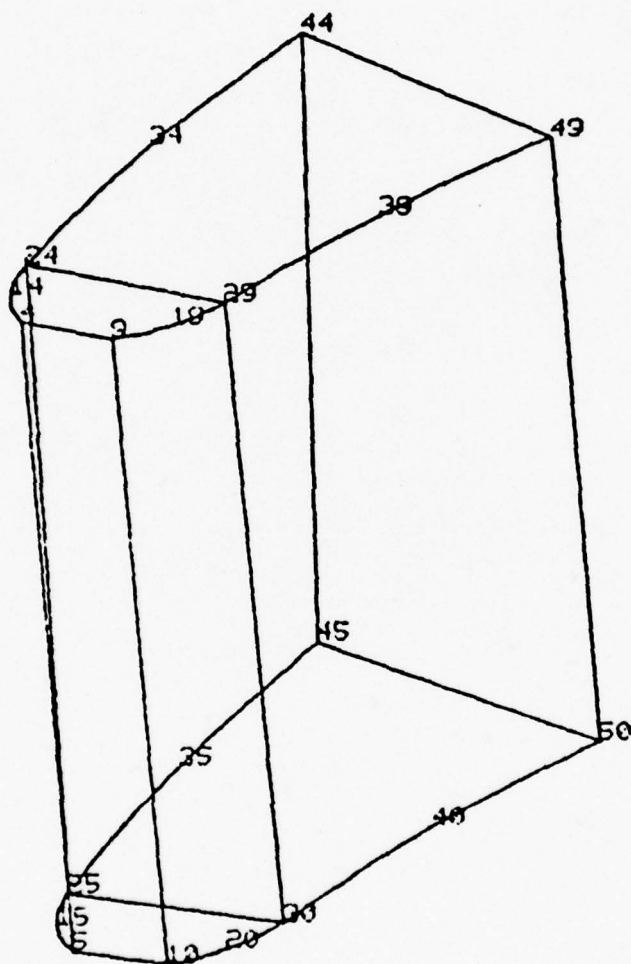


Figure 21. 12-Node Airfoil Section Showing Truncation of Leading Edge Element

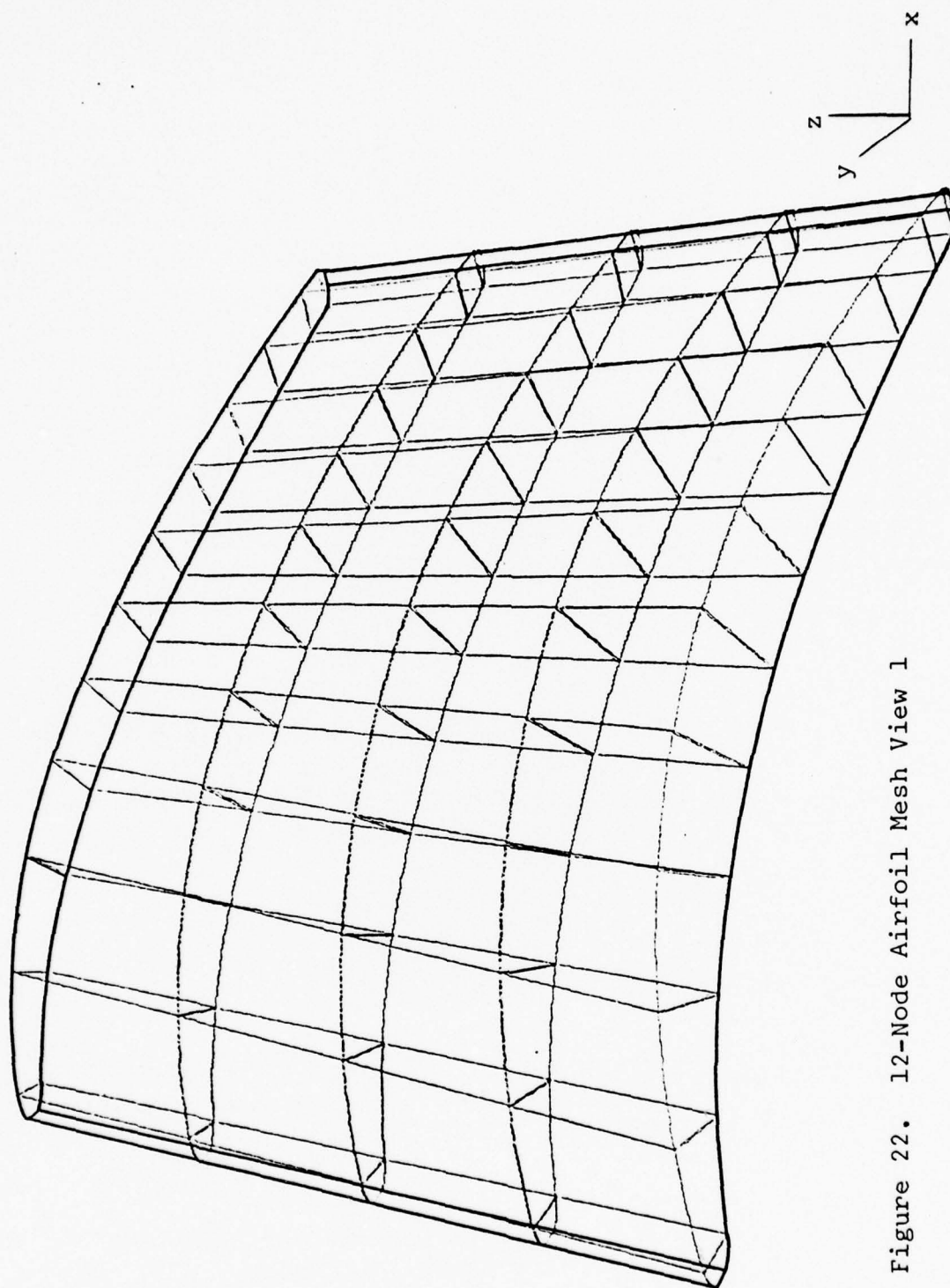


Figure 22. 12-Node Airfoil Mesh View 1

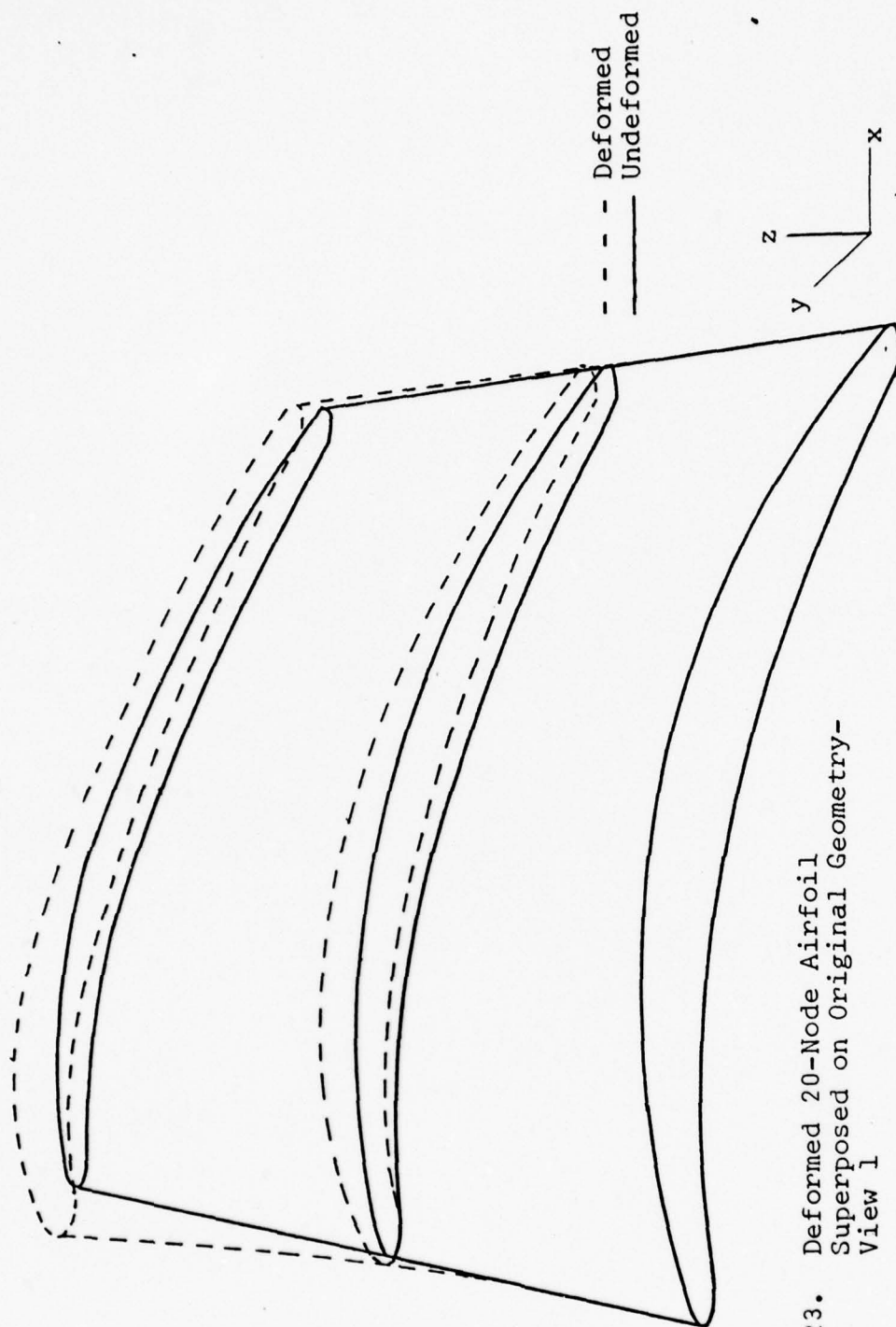


Figure 23. Deformed 20-Node Airfoil
Superposed on Original Geometry-
View 1

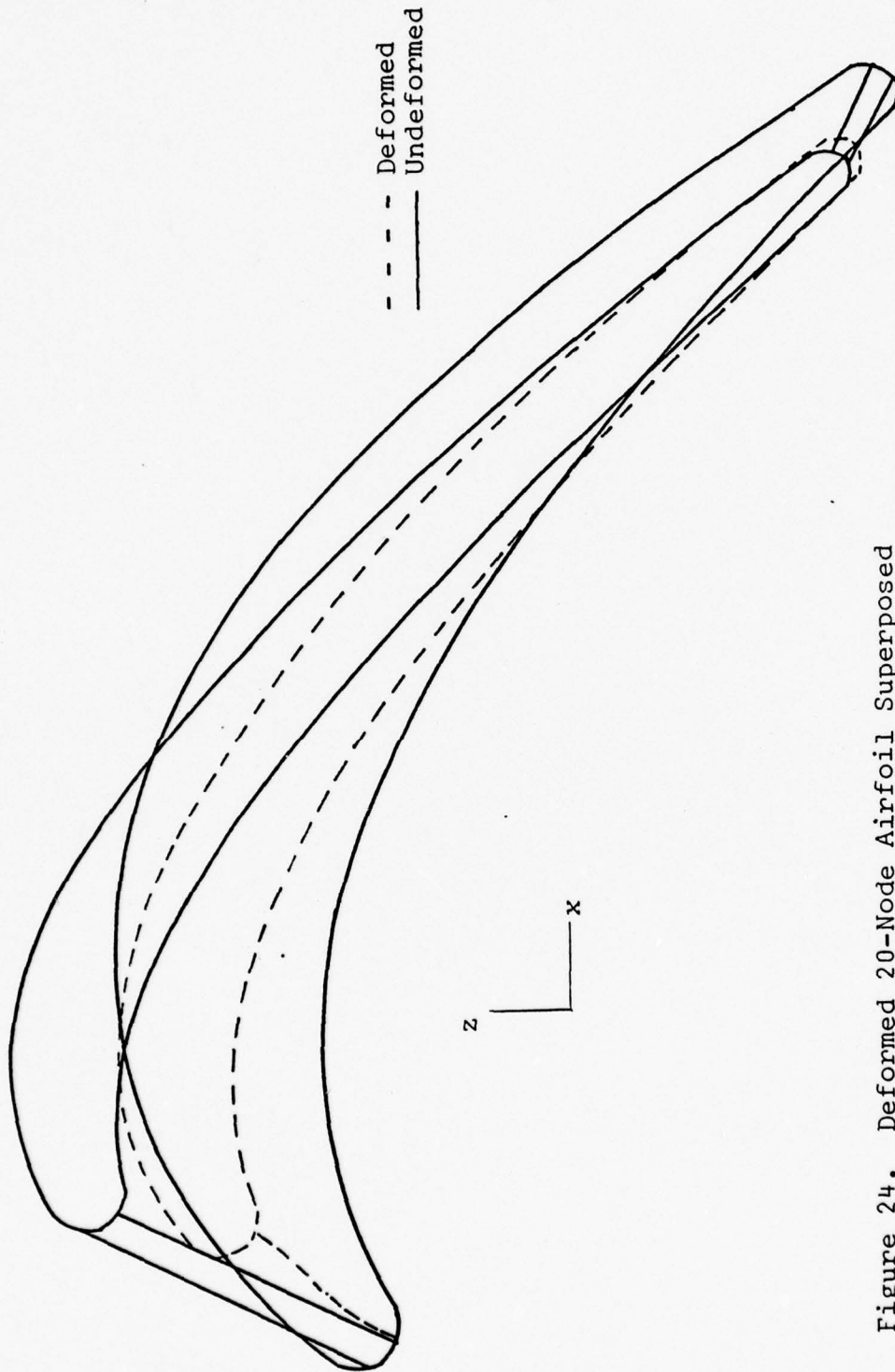


Figure 24. Deformed 20-Node Airfoil Superposed on Original Geometry-View 2

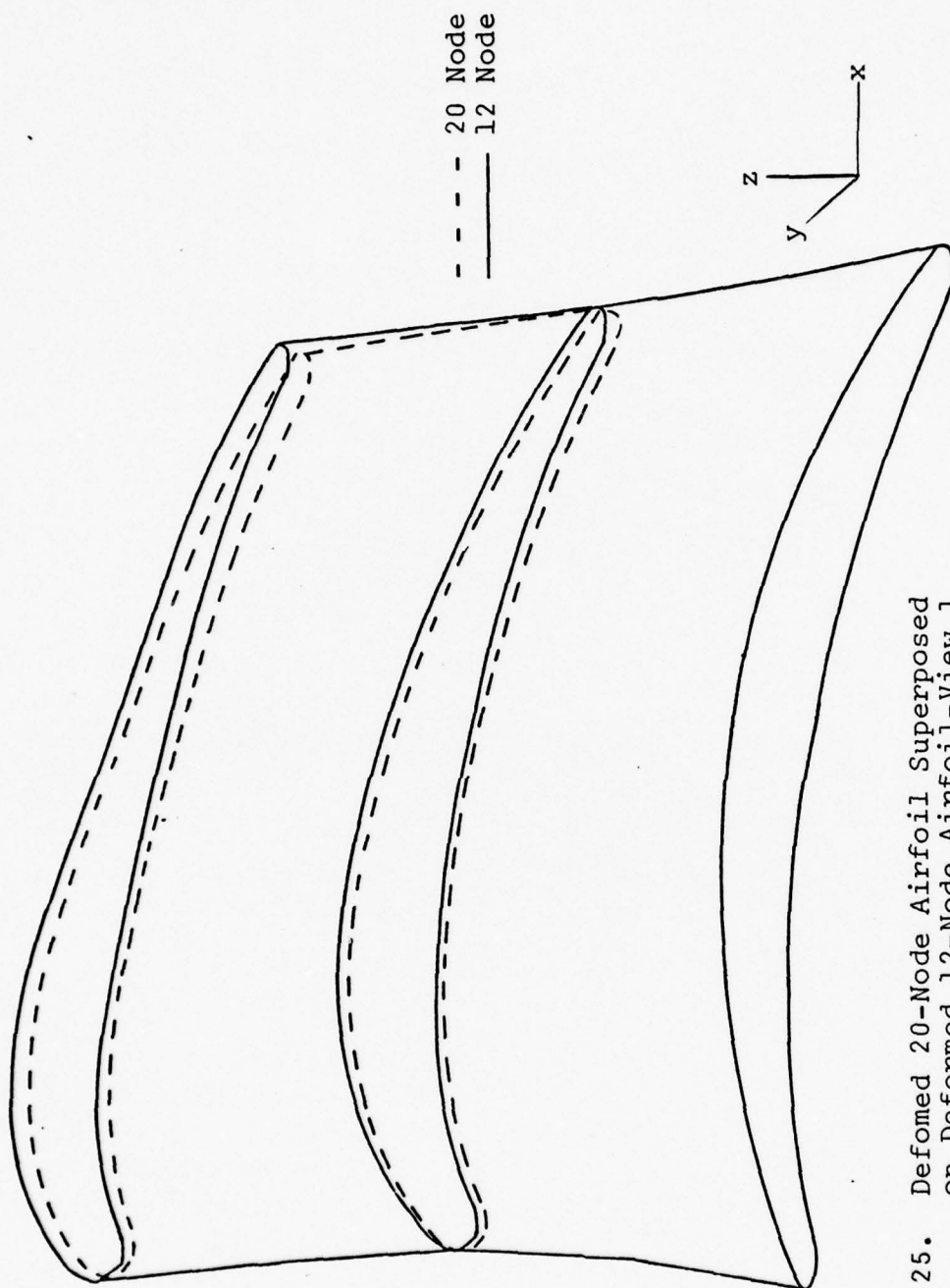


Figure 25. Deformed 20-Node Airfoil Superposed on Deformed 12-Node Airfoil-View 1

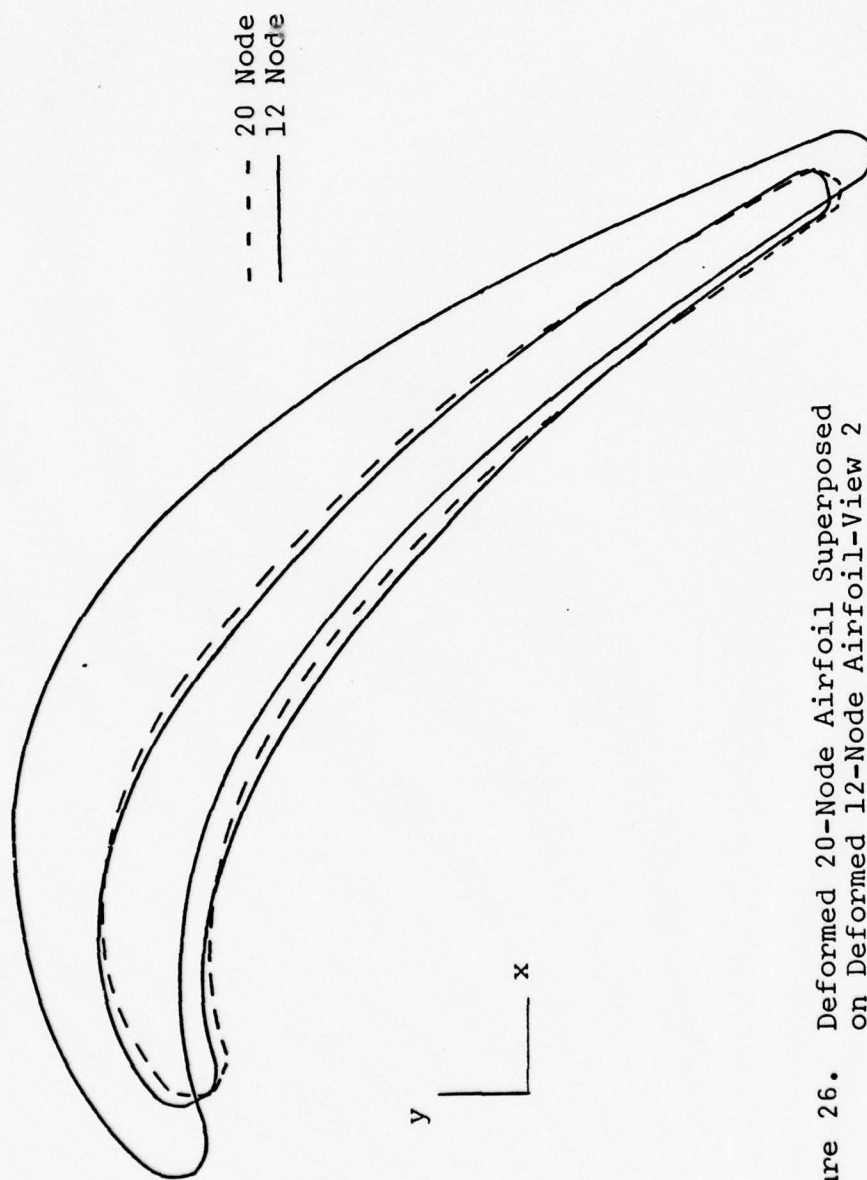


Figure 26. Deformed 20-Node Airfoil Superposed on Deformed 12-Node Airfoil-View 2

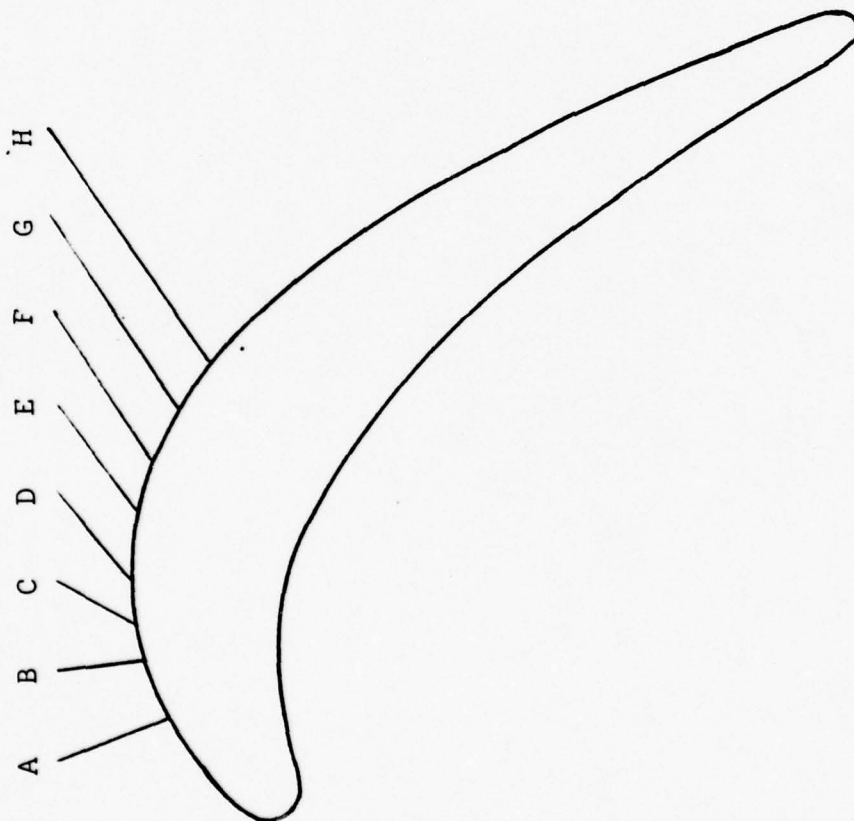


Figure 27. Airfoil Root Section Showing Location of Nodal Points Used for Comparison in Tables II and III

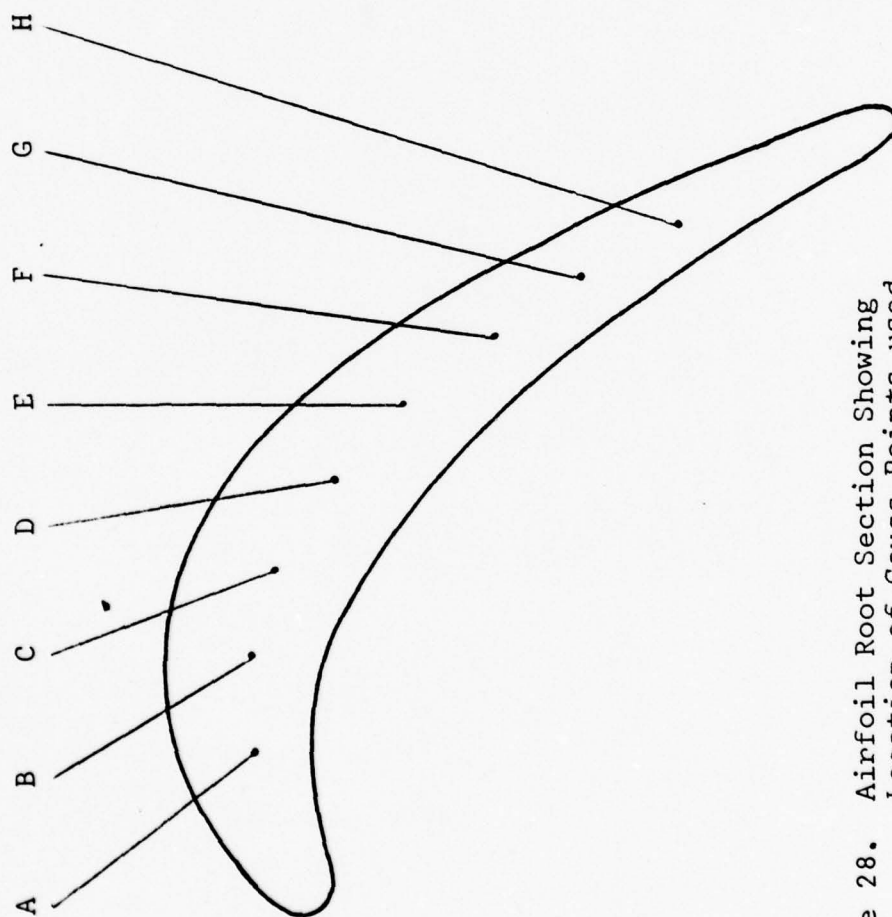


Figure 28. Airfoil Root Section Showing
Location of Gauss Points used
for Comparison in Table IV

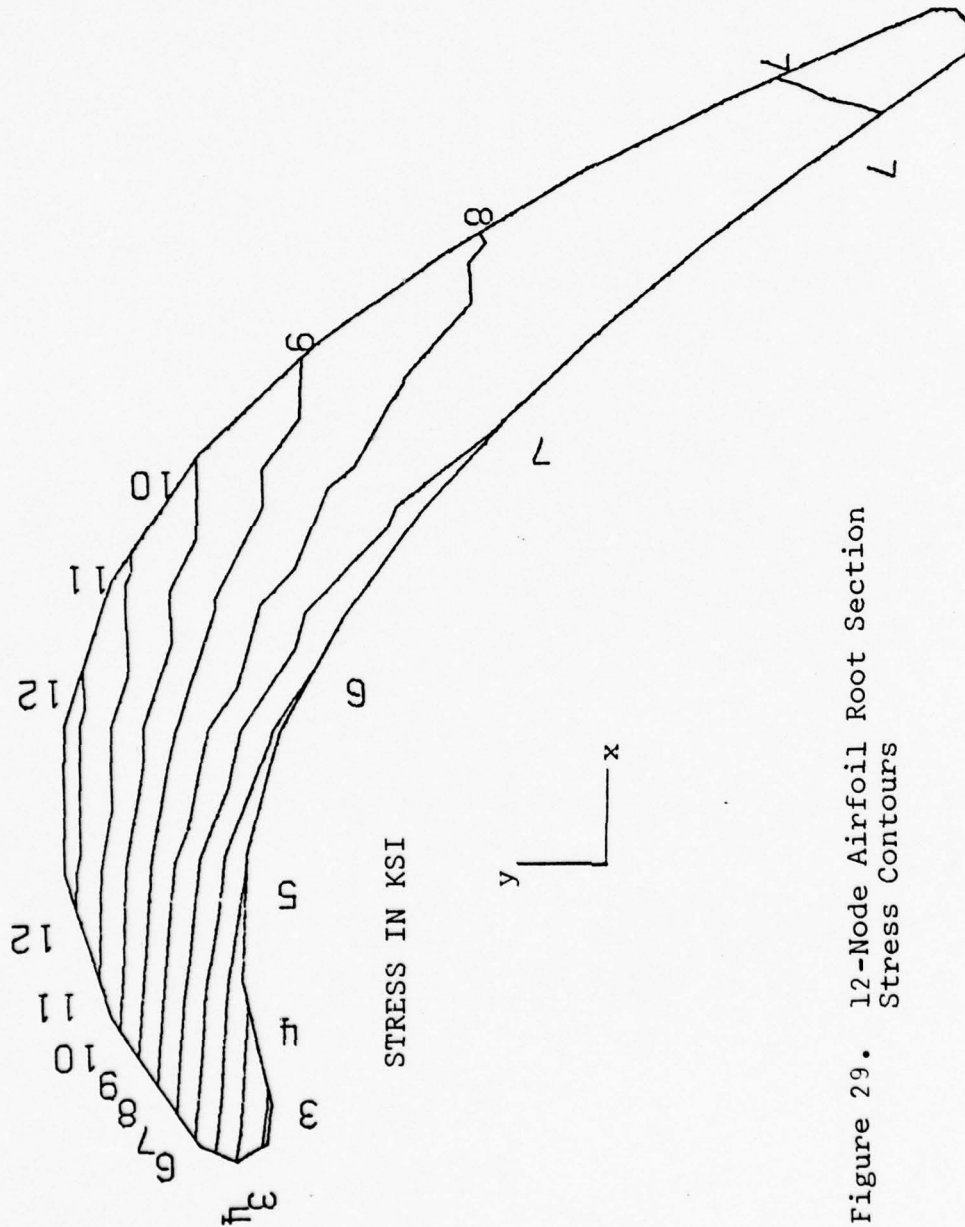


Figure 29. 12-Node Airfoil Root Section Stress Contours

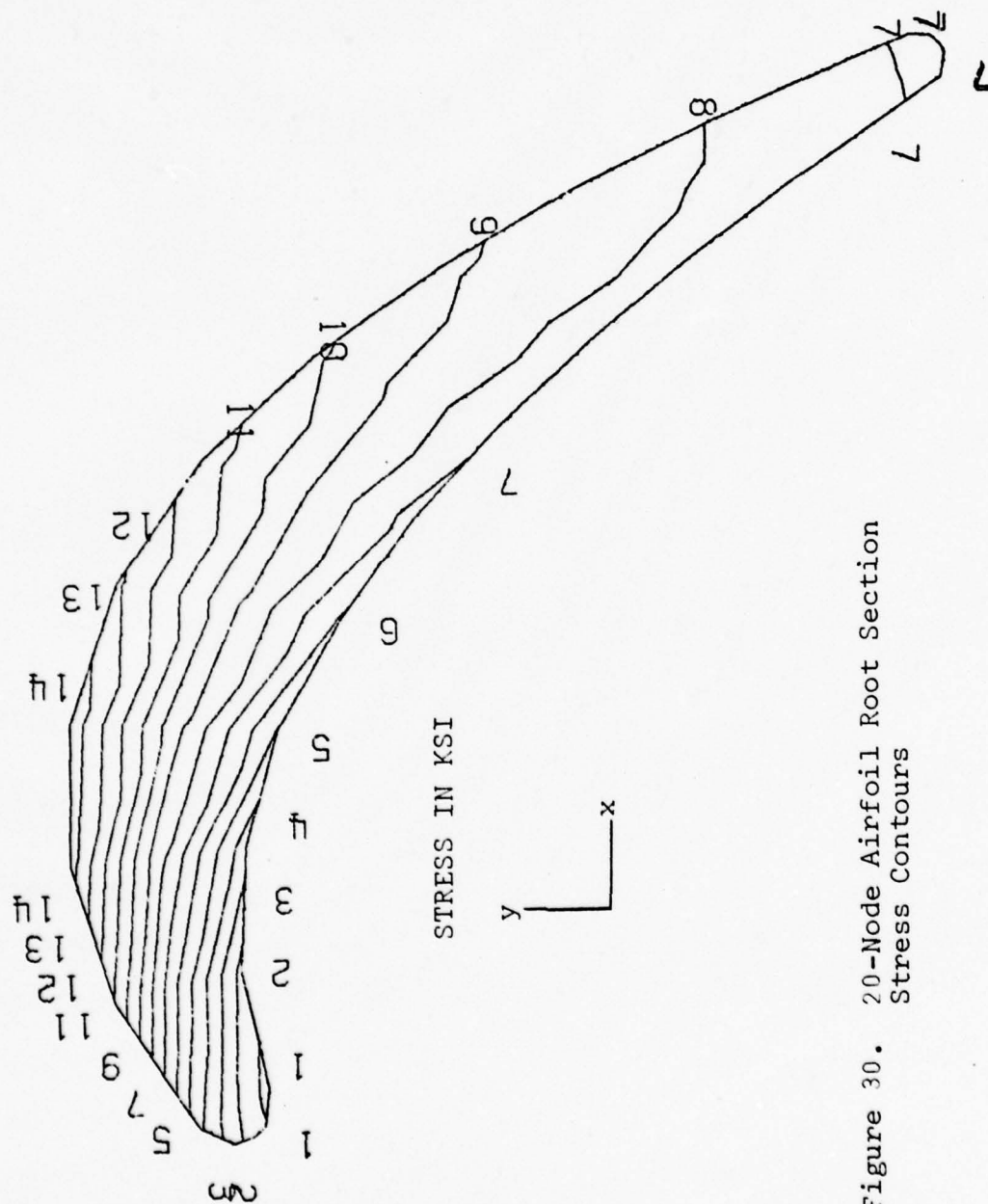


Figure 30. 20-Node Airfoil Root Section Stress Contours

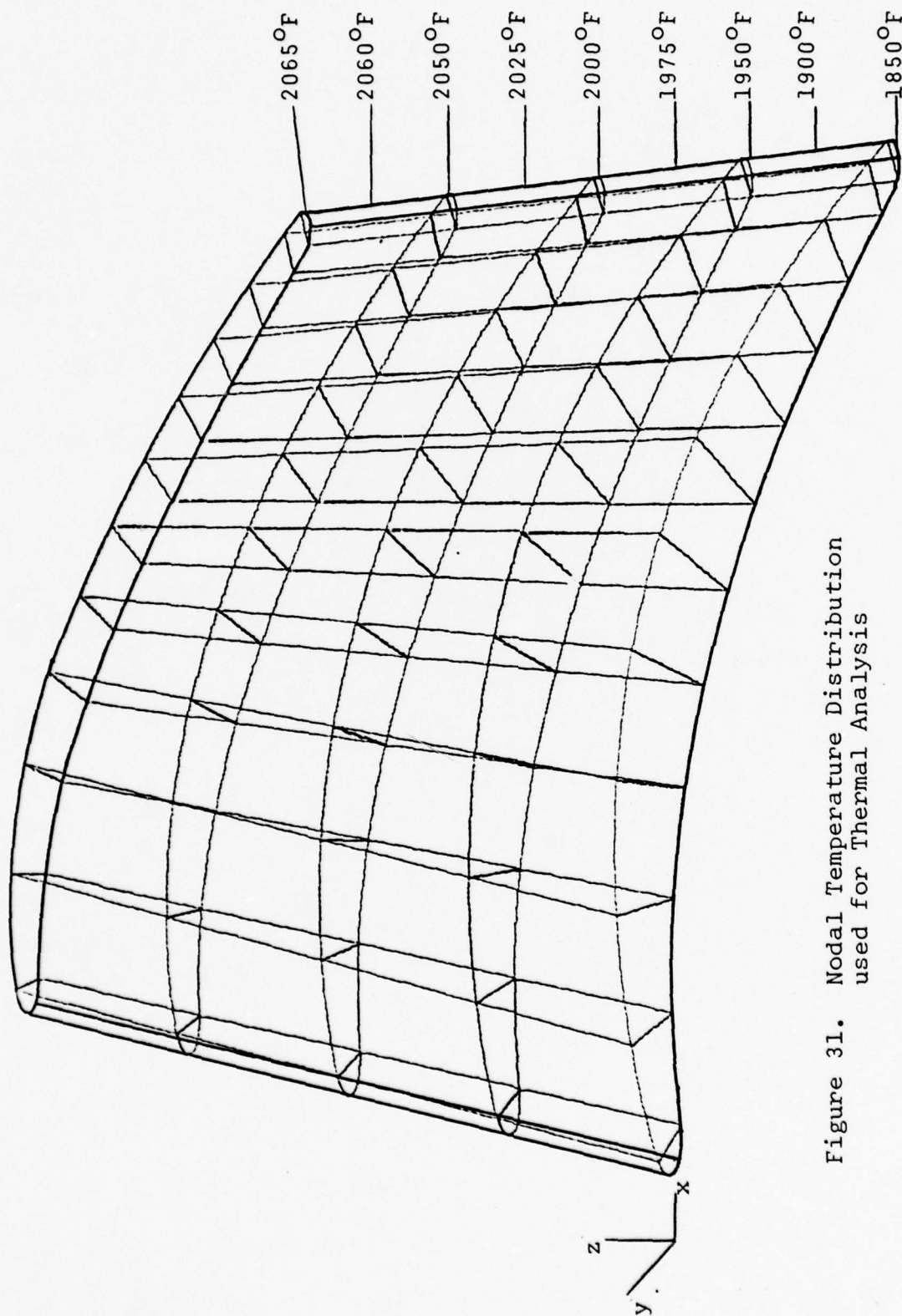


Figure 31. Nodal Temperature Distribution
used for Thermal Analysis

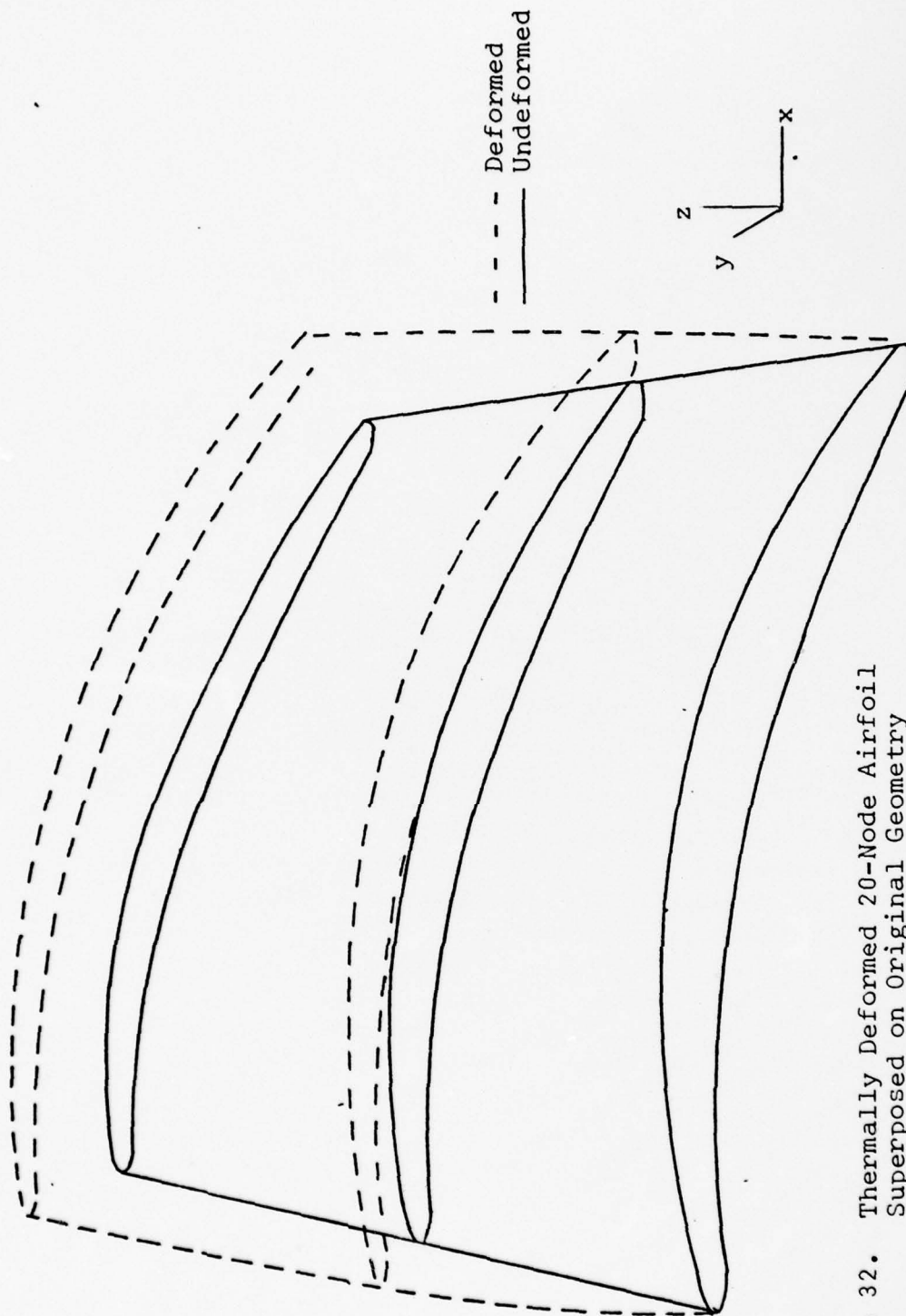


Figure 32. Thermally Deformed 20-Node Airfoil
Superposed on Original Geometry

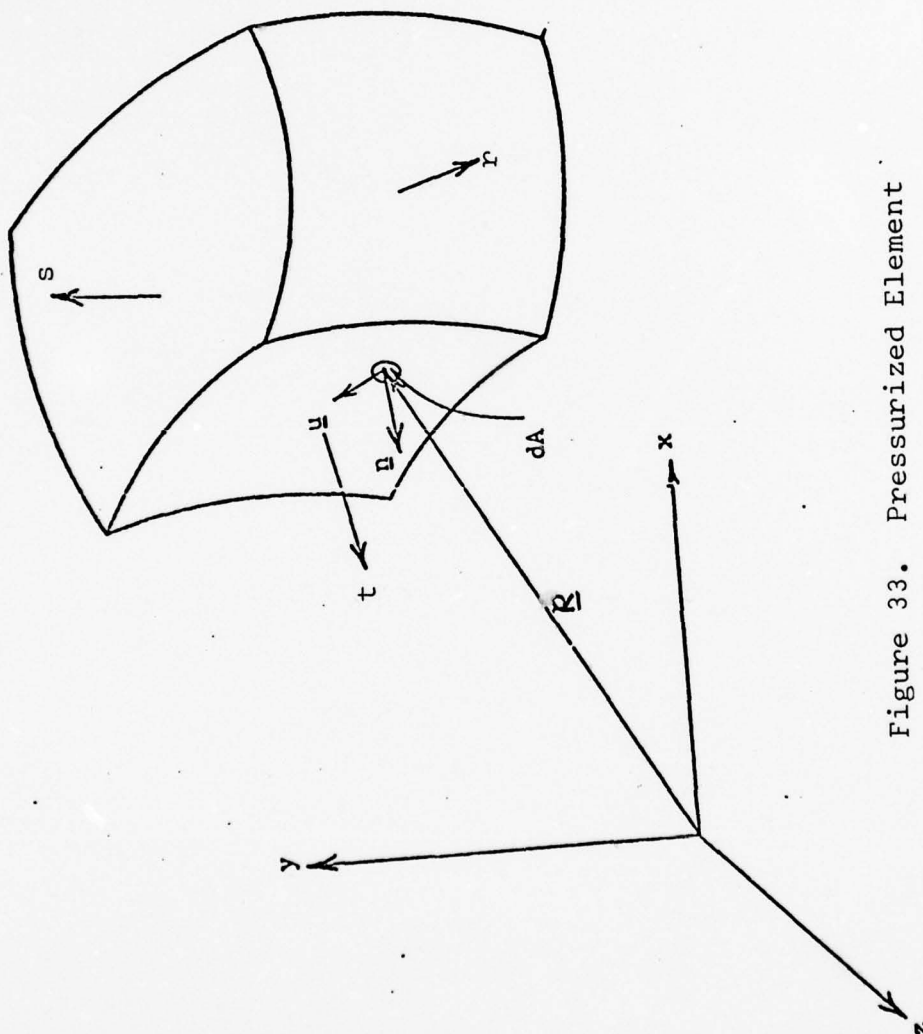


Figure 33. Pressurized Element

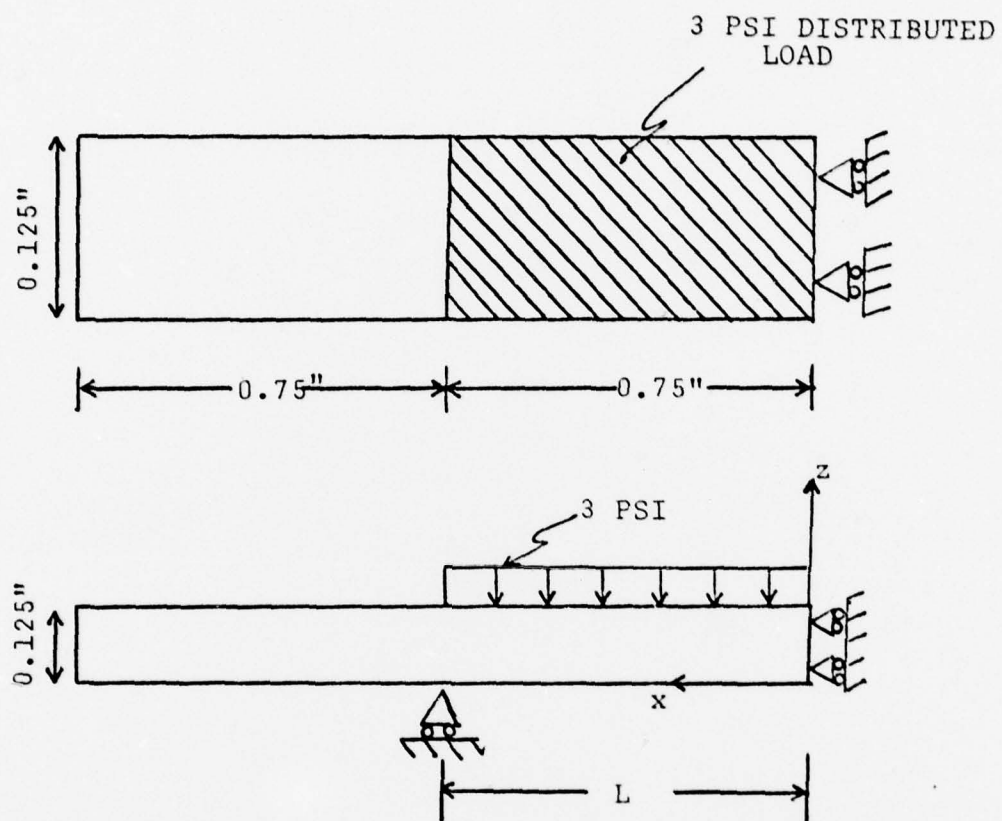


Figure 34. Pressure Load Verification Problem

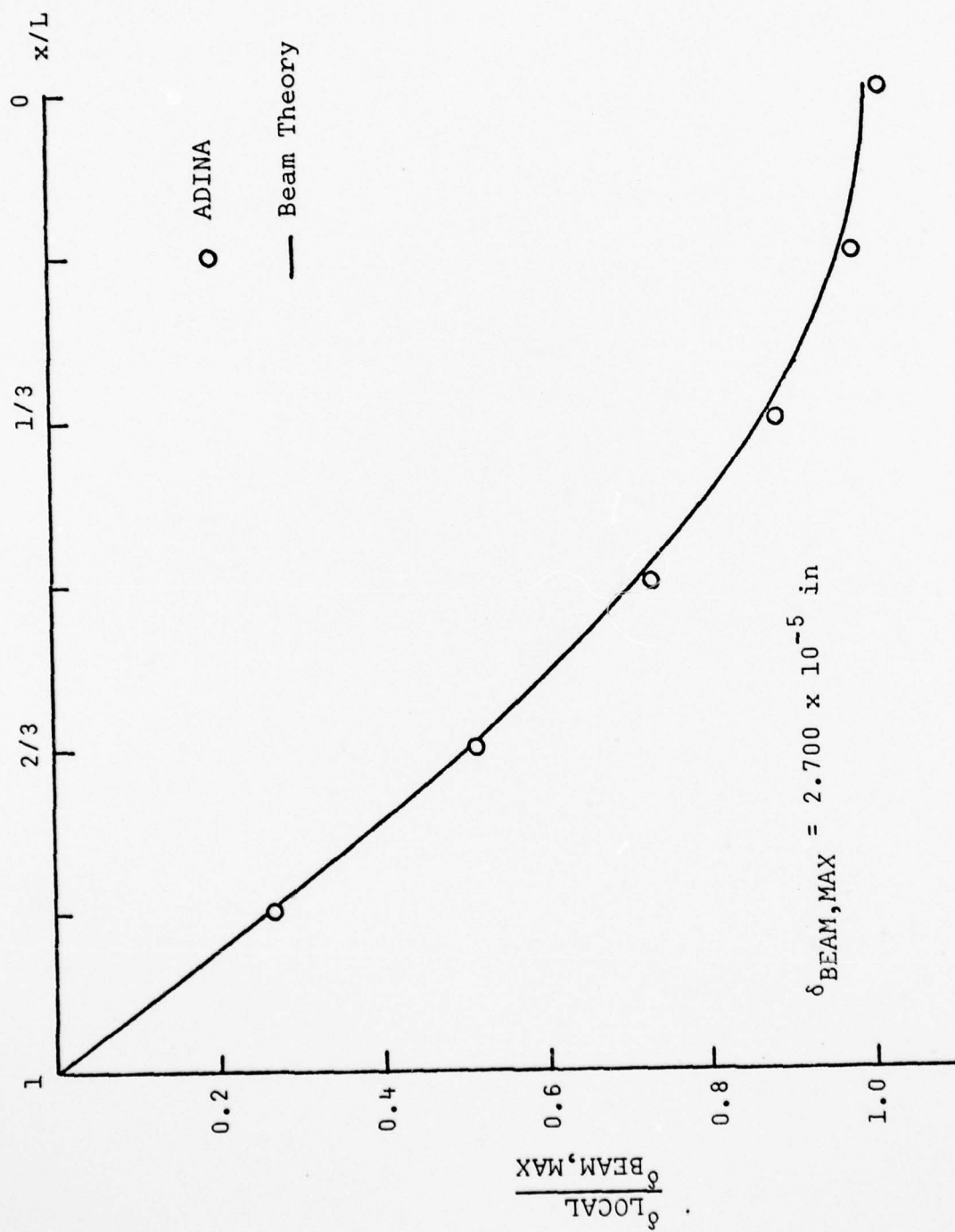


Figure 35. Normalized Displacement vs. Span for Bar Problem with Pressure Load

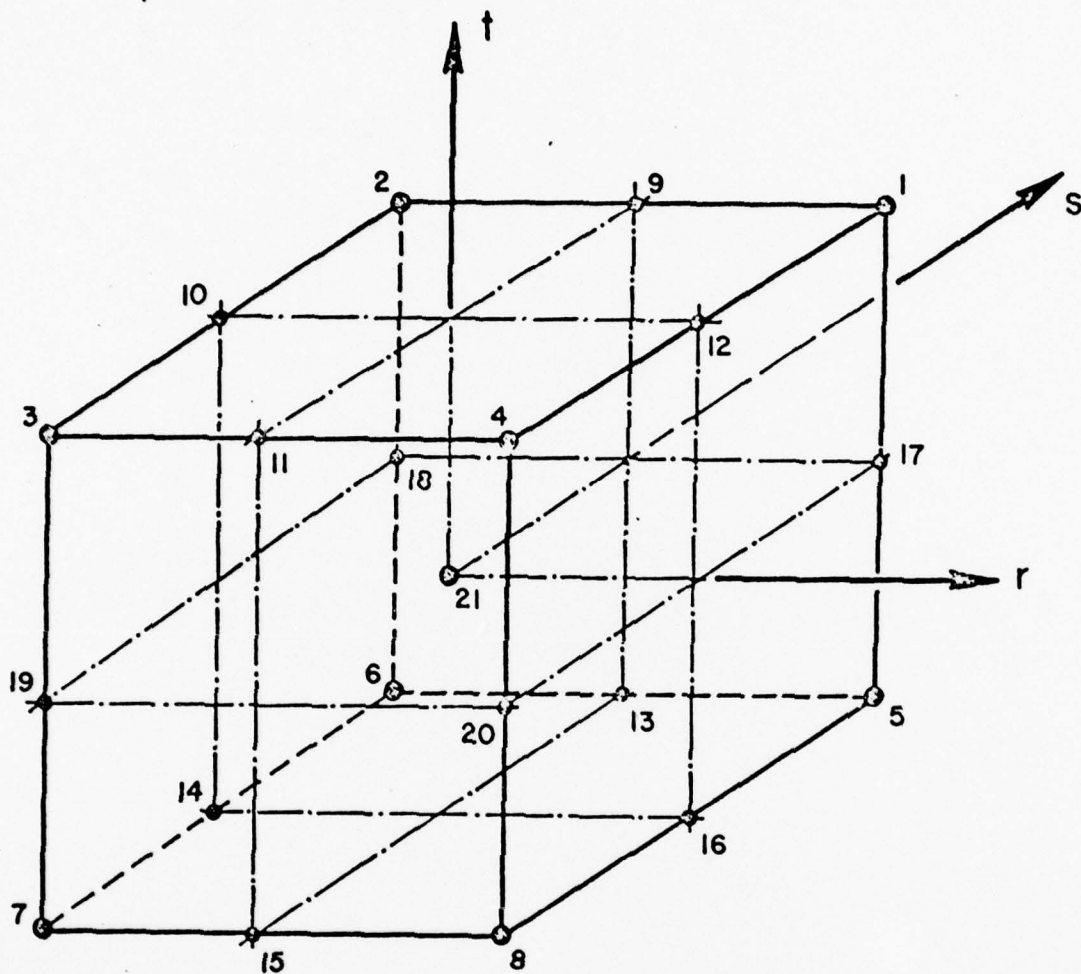


Figure 36. Hexahedral Element in Natural Coordinates

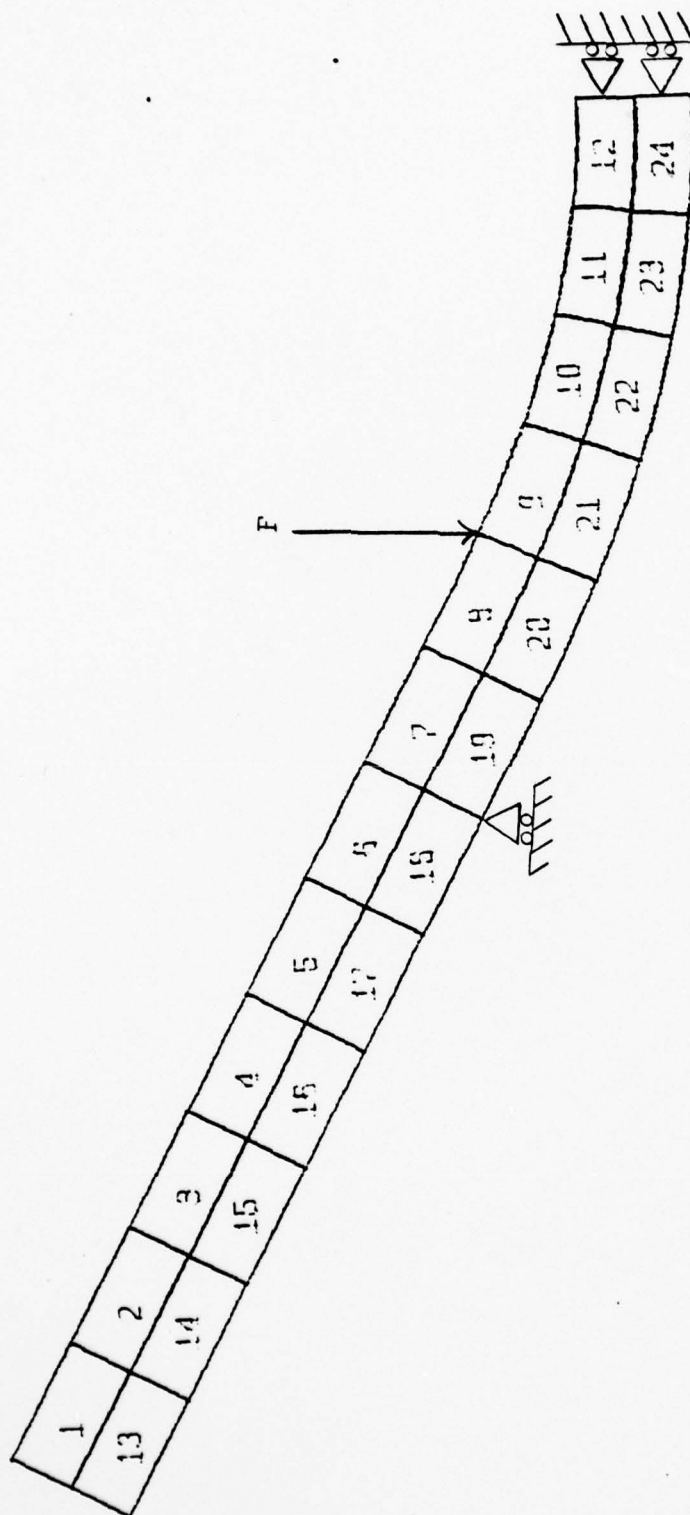


Figure 37. Deformed Structure for Flexure Problem

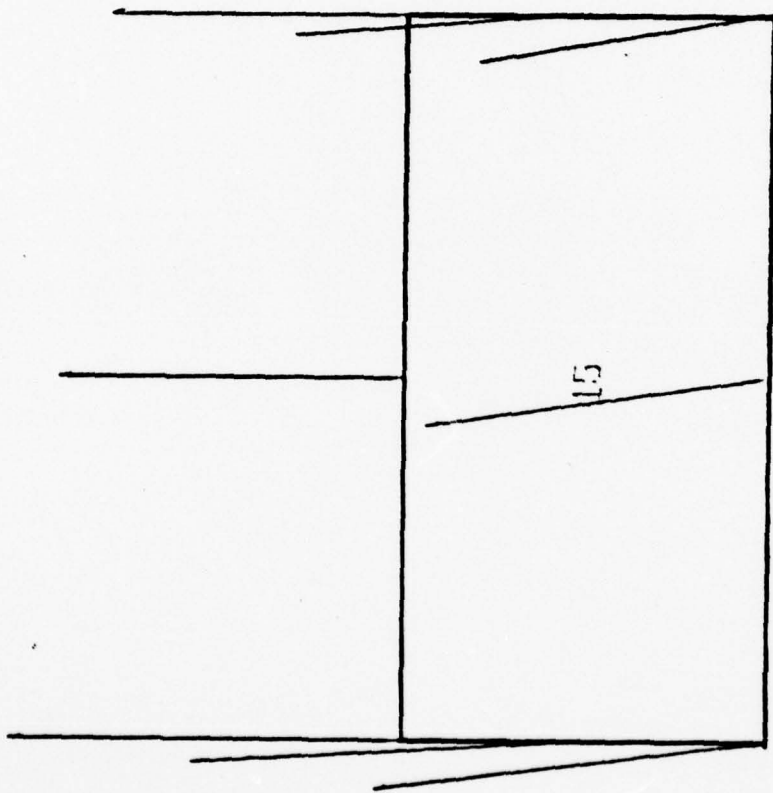


Figure 38. Postprocessing Using Displacement Vectors
at the Nodes (Flexure Problem)

TABLE I: MESH CHARACTERISTICS

20-NODE BRICKS

Element type.....	20-node brick
Number of nodes.....	452
Number of elements.....	52
Degrees of freedom.....	1152
Mean half-bandwidth.....	93
Maximum half-bandwidth.....	129
Number of stiffness matrix elements.....	107064

12-NODE BRICKS

Element type.....	12-node brick
Number of nodes.....	270
Number of elements.....	52
Degrees of freedom.....	648
Mean half-bandwidth.....	45
Maximum half-bandwidth.....	66
Number of stiffness matrix elements.....	29106

TABLE II. COMPARISON OF MAXIMUM STRESSES IN AIRFOIL ROOT SECTION (Cf. Fig. 27)

LOCATION	FOIL	σ_{xx} (PSI)	σ_{yy} (PSI)	σ_{zz} (PSI)
A	12	2553	2553	10213
	20	2658	2658	10634
B	12	2897	2897	11588
	20	3211	3211	12843
C	12	3082	3082	12329
	20	3560	3560	14244
D	12	3130	3130	12520
	20	3656	3656	14626
E	12	3074	3074	12300
	20	3625	3625	15000
F	12	2946	2946	11784
	20	3447	3447	13788
G	12	2781	2781	11125
	20	3231	3231	12924
H	12	2608	2608	10432
	20	3000	3000	12001

TABLE III. COMPARISON OF MAXIMUM PRINCIPAL STRESSES IN AIRFOIL ROOT SECTION (cf. Fig. 27)

LOCATION	σ_1 12-NODE AIRFOIL (PSI)	σ_1 20-NODE AIRFOIL (PSI)	% DIFFERENCE
A	10441	10865	3.9
B	11790	12992	9.25
C	12519	14379	12.9
D	12706	14759	13.9
E	12486	14676	14.9
F	11969	13998	14.5
G	11308	13192	14.3
H	10608	12293	13.7

$$\sigma_1 > \sigma_2 > \sigma_3$$

$$\% \text{ Difference} \equiv \frac{\sigma_1(20) - \sigma_1(12)}{\sigma_1(20)}$$

TABLE IV. COMPARISON OF SELECTED THERMAL STRESSES IN
AIRFOIL ROOT SECTION (cf. Fig. 28)

LOCATION	σ_{zz} 12 NODE (PSI)	σ_{zz} 20 NODE (PSI)	% DIFF
A	-11590	-11407	-1.6
B	-7744	-8664	10.6
C	-4387	-5759	23.8
D	-1813	-3487	48.0
E	-364	-2337	84.0
F	169	-1800	109.4
G	-668	-2431	72.5
H	-3560	-4998	28.8
I	-10543	-12554	16.01

$$\% \text{ Difference} \equiv \frac{\sigma(20) - \sigma(12)}{\sigma(20)}$$

TABLE V. FACE NUMBERING CONVENTION FOR PRESSURE LOADS

FACE NUMBER	NATURAL COORDIN.	CORNER NODES N ₁ N ₂ N ₃ N ₄	MIDSIDE NODES N ₁ N ₂ N ₃ N ₄	α	β
1	(+l, s, t)	1 4 8 5	12 20 16 17	s	t
2	(-l, s, t)	2 3 7 6	10 19 14 18	t	s
3	(r, +l, t)	1 5 6 2	17 13 18 9	t	r
4	(r, -l, t)	4 8 7 3	20 15 19 11	r	t
5	(r, s, +l)	1 2 3 4	9 10 11 12	r	s
6	(r, s, -l)	5 6 7 8	13 14 15 16	s	r

APPENDIX A: PREPARATION OF TEMPERATURE TAPE

If temperature input is used with the ADINA program, nodal point temperatures are stored in sequence on device number 56 [4]. At NPS, ITEL 3330 disks have been used, although any storage device may be designated. $NSTE + 1$ records are read, where $NSTE$ is the number of solution steps. (In linear static analysis, $NSTE$ usually equals 1.)

The read statement in the program is:

```
READ (56) TIME, (TEMPV(I), I=1, NUMNP).
```

On each record, the problem time of the nodal temperatures is read, followed by an implicit "do-loop" from 1 to the total number of node points ($NUMNP$) to read each nodal point temperature.

A program to establish the temperature tape is listed at the end of this Appendix. It will be discussed in three parts: (A) job control language, (B) the `RDADIN` subroutine and (C) the binary write.

A. TEMPERATURE TAPE JOB CONTROL LANGUAGE (JCL)

The program listing in Appendix B starts with a two step procedure [17]. The first is a purge operation that returns the previously reserved disk space to the operating system, the second step reserves disk space for the data set which is named `F0099.TAPE`. One cylinder of 3330 disk space (248K bytes) is reserved on one of the NPS computer center resident

disks. The data set has a variable block size, and an expiration date of 360 days after establishment.

The JCL employed combines ease of continued use of the disk space, with minimum chances of data set expiration.

B. THE RDADIN SUBROUTINE

The RDADIN subroutine was originally programmed as part of the PSAP1 preprocessor. The subroutine will read an ADINA input deck from title card through element connectivity. It generates node numbers, nodal coordinates, element numbers and element connectivities for the entire mesh. Each of the above is stored in a linear array. The programmer is then able to utilize any or all of these arrays to input the nodal point temperature.

For example, in the linear airfoil mesh there were isothermal planes. The matrix of z-coordinates (radial distance from axis of rotation) was used. Each z-coordinate was checked by a series of 'if statements', and nodal temperatures were assigned.

C. THE BINARY WRITE

After the matrix of nodal temperatures was established, the next step was to write it sequentially onto device 56. Array RCD1 contained the problem time for the temperatures, and temperatures for each node; i.e., it was a linear array of dimension NUMNP + 1. This array could then be written onto device 56 by the statement:

```
WRITE (56)RCD1.
```

Device 56 must then be defined in the GO step by:

```
//GO.FT56F001 DD UNIT=3330,VOL=SER-DISK01,  
//^DSN=F0099.TAPE,DISP=OLD
```

Each time the WRITE statement is executed one record is established. The sequence must be repeated until $NSTE + 1$ records are established, where NSTE is the number of solution steps.

In the linear airfoil problem, there was only one solution time step. Therefore, only two records on device 56 were needed. One record was at $time=0$, and the next record was one time step increment later.

It is recommended that a hard copy of the RCD1 matrix be printed to ascertain the accuracy of the temperature tape.

AD-A055 538

NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF
STRESS ANALYSIS OF CERAMIC GAS TURBINE BLADES BY THE FINITE ELE--ETC(U)
MAR 78 J H PREISEL

F/G 21/5

UNCLASSIFIED

NPS-69-78-010-PT-2

NL

2 of 2

AD
A055 538



APPENDIX B. NODAL TEMPERATURE TAPE PROGRAM

```

// (STANDARD GREEN JOB CARD)
// EXEC PGM=IEHPRGM
// SYSPRINT DD SYSOUT=A
// DISK DD UNIT=3330,VOL=SER=DISK01,DISP=SHR
// SYSIN DD *
// SCRATCH VOL=3330=DISK01,DSNAME=F0099.TAPE,PURGE
/*
// EXEC PGM=IEFBR14
// TEMP DD DSN=F0099.TAPE,UNIT=3330,VOL=SER=DISK01,DISP=(NEW,KEEP),
// LABEL=RETPD=360,SPACE=(CYL,(1,1,1))
/*
// EXEC FORTCLG
// FORT:SYSDIN DD *
// FORT:IMPLICIT REAL*8(A-H,O-Z)
*****00000010*****
C

```

THIS PROGRAM CAN BE USED TO ESTABLISH A TEMPERATURE TAPE FOR THE ADINA CODE. THE READIN SUBROUTINE GENERATES THE XPT,YPT,ZPT MATRICES OF COORDINATES. ALSO, THE CONNECTIVITY MATRIX (INCONT), THE ELEMENT NUMBERS AND NODAL NUMBERS ARE STORED IN MATRIX FORM. (INREL AND NUMPT). ANY OF THESE MATRICES CAN BE USED WITH IF STATEMENTS TO ESTABLISH NODAL CONNECTIVITY. THE SUBROUTINE TEMPRT IS PRESENTLY SET UP TO CHECK AGAINST THE Z COORDINATES.

NODAL TEMPERATURES ARE WRITTEN SEQUENTIALLY ON DEVICE 58

```

*****
COMMON A(15000)
COMMON/IPARA/NUMNP,NUMEL
NUMNP=270
NUMEL=52
MTOT=15000
N1=1
N2=N1+NUMNP+7*NUMEL
N3=N2+NUMNP+7*NUMEL
N4=N3+NUMNP+7*NUMEL
N5=N4+20*NUMEL
N6=N5+27*NUMEL
N7=N6+NUMNP+7*NUMEL
N8=N7+NUMEL
N9=N8+6*NUMEL
N10=N9+6*NUMEL
*****
00000020
00000030
*****
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160

```



```

NLASt=N10+6*NUMEL
IF(NLASt.GT.MTOT)GO TO 10
CALL READIN(A(N1),A(N2),A(N3),A(N4),A(N6),A(N7))
CALL TEMPRT(A(N3),A(N1))
STOP
10 WRITE(6,1000)NLASt
1000 FORMAT(//5X,'MTOT SHOULD BE INCREASED TO A VALUE OF ',I5//)
STOP
SUBROUTINE READIN(XPT,YPT,ZPT,NCONT,NREL,NUMPT)
*****
SUBROUTINE TO READ ADINA DECK.
*****
IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 TITLE(20)
COMMON/IPARA/NUMNP,NUMEL
COMMON/LOADS/NLOAD,NLCUR,NPTM,IDGRAV
COMMON/SECTION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCONT(20,1)
DIMENSION IDOF(6),ID(6),IDOLD(6),NCDE(20),NPAR(20)
1 ,NP(20),INP(20)
DATA CTEST/'X'
NCARD=0
READ(5,9000)(TITLE(I),I=1,20)
WRITE(6,9011)(TITLE(I),I=1,20)
FORMAT(//5X,20A4,///)
9011 READ MASTER CONTROL CARDS
C *** NUMNP = TOTAL NUMBER OF NODE POINTS
C *** NELTYP = NUMBER OF ELEMENT GROUPS
9001 READ(5,9001) NUMNP, IDOF(1),I=1,6),NEGL,NEGNL,MODEX,NSTE
FORMAT(15,6I1,14,3I5)
NELTYP=NEGL+NEGNL
NNODE=NUMNP
READ(5,7002) IMASS,IDAMP,IMASSN,ICAMPN
7002 FORMAT(4I5)
READ(5,7002) IEIG
READ(5,7002) ISREF,NUMREF,IEQUIT,ITEMAX
READ(5,9000) DUMMY
READ(5,9000) DUMMY
READ(5,9000) DUMMY
C *** READ OR GENERATE NODAL POINT DATA
NULD=0
NEQ=0
10 READ(5,9006)CT,N, ID(1),I=1,6),XPT(N),YPT(N),ZPT(N),KN
9006 FORMAT(A1,I4,I4,I4,5I5,3F10.0,I5)

```



```

C *** CHECK FOR CYLINDRICAL COORDINATES
IF(CT-NE-CTEST) GO TO 12
DUM=ZPT(N)/57.2958
R=YPT(N)
YPT(N)=R*DCOS(ZPT(N)/57.295800)
ZPT(N)=R*DSIN(ZPT(N)/57.295800)
12 CONTINUE
NUMPT(N)=N
IF(NOLD.EQ.0) GO TO 53
FOR GENERATION OF FIXED BOUNDARY CONDITIONS
CC 15 I=1,6
IF(IDOLD(I).EQ.-1.AND.ID(I).EQ.0) ID(I)=IDOLD(I)
15 CONTINUE
IF(KNOLD.EQ.0) GO TO 50
NUM=(N-NOLD)/KNOLD
NUMN=NUM-1
IF(NUMN.LT.1) GO TO 53
TO COUNT DOFS TO DETERMINE NUMBER OF IC CARCS
CC 20 I=1,6
IF(IDOF(I).EQ.0.AND.IDOLD(I).EQ.0) NEQ=NEQ+NUMN
20 CONTINUE
DX=(XPT(N)-XPT(NOLD))/NUM
IF(CT-NE-CTEST) GO TO 21
RCLD=YPT(NOLD)/DCOS(DUMOLD)
RNEW=YPT(N)/DCOS(DUM)
DR=(RNEW-RCLD)/NUM
DT=(DUM-DUMOLD)/NUM
GC TO 22
21 CONTINUE
DY=(YPT(N)-YPT(NOLD))/NUM
DZ=(ZPT(N)-ZPT(NOLD))/NUM
22 CCNTINUE
K=NOLD
DC 30 J=1,NUMN
KK=K
K=K+KNOLD
XPT(K)=XPT(KK)+OX
IF(CT-NE-CTEST) GO TO 26
ROLD=RCLD+DR
DUMOLD=DUMOLD+DT
YPT(K)=ROLD*DCOS(DUMOLD)
ZPT(K)=ROLD*DSIN(DUMOLD)
GO TO 28
26 CONTINUE
YPT(K)=YPT(KK)+DY
ZPT(K)=ZPT(KK)+DZ
CONTINUE
NUMPT(K)=K
28

```

```

00002060
00002070
00002080
00002090
00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002170
00002180
00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260
00002270
00002280
00002290
00002300
00002310
00002320
00002330
00002340
00002350
00002360
00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530

```

00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950
00002960
00002970
00002980
00002990
00003000
00003010

```

30      CONTINUE
50      NCLO=N
      KNOLD=KN
      DUMOLD=DUM
      TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS
C ***  DO 55 I=1,6
      IF(100F(I).EQ.0.AND.ID(I).EQ.C) NEQ=NEQ+1
      IDOLD(I)=ID(I)
      CONTINUE
55      IF(N-NE.NUMPI) GO TO 10
      READ LOAD CONTROL CARDS
      READ(5,9012)NLOAD,NLCUR,NPTM,IOGRAV
9012    FCRMAT(415)
      DO 80 I=1,IMASSN
      IF(IMASSN.EQ.0) GO TO 81
      READ(5,9000) DUMMY
      CONTINUE
80      CONTINUE
81      IF(IDAMPN.EQ.0) GO TO 91
      DO 90 I=1,IDAMPN
      READ(5,9000) DUMMY
      CONTINUE
90      CONTINUE
91      READ INITIAL CONDITIONS
      READ(5,9002) ICON
      IF(ICON.EQ.0) GO TO 100
      CARDNR=NEQ/6.0
      NCARD=IDINT(CARDNR)
      TEST=CARDNR-NCARD
      IF(TEST.GT.0.1) NCARD=NCARD+1
      DO 95 I=1,NCARD
      READ(5,9000) DUMMY
      CONTINUE
95      IF(IMASS.EQ.0) GO TO 100
      DO 96 I=1,NCARD
      READ(5,9000) DUMMY
      CONTINUE
96      DO 98 I=1,NCARD
      READ(5,9000) DUMMY
      CONTINUE
98      FCRMAT(6E12.6)
9007    CONTINUE
100      NUMEL=0
      WRITE(6,9009) NEQ,NCARD
9009    FORMAT(//,' NEQ AND NCARD FOR IC IN GEOM1 = ',15,10X,15//)
C ***  READ ELEMENT CONTROL CARDS
      DO 900 M=1,NELTYP
      READ(5,9008)(NPAR(I),I=1,20)

```

```

9008 WRITE(6,901C) (NPAR(I),I=1,20)
9010 FCRMAT(2014)
      MTYPE=NPAR(16)
      NSTRES=NPAR(13)
C *** CALCULATE THE NUMBER OF MATERIAL CASE CARDS
      IF(NPAR(15).EQ.1) NCARD=1
      IF(NPAR(15).EQ.2) NCARD=2+NPAR(18)
      IF(NPAR(15).EQ.3) NCARD=4
      IF(NPAR(15).EQ.4) NCARD=4
      IF(NPAR(15).EQ.5) NCARD=2
      IF(NPAR(15).EQ.8) NCARD=1
      IF(NPAR(15).EQ.9) NCARD=1
      IF(NPAR(15).EQ.10) NCARD=6
      IF(NPAR(15).EQ.11) NCARD=6
      IF(NPAR(15).NE.12) GO TO 111
      CARDNR=NPAR(17)/8.0
      NCARD=IDINT(CARDNR)
      TEST=CARDNR-NCARD
      IF(TEST.GT.0.1) NCARD=NCARD+1
      CCNTINUE
111 N20=20
C *** READ MATERIAL PROPERTIES
      DC 222 J=1,NUMMAT
      READ(5,9000) DUMMY
9000 FORMAT(20A4)
      DO 45 I=1,NCARD
45 READ(5,9000) DUMMY
      CCNTINUE
222 READ STRESS OUTPUT TABLE CARDS
      IF(NPAR(13).EQ.0) GO TO 61
      DC 60 I=1,NSTRES
      READ(5,9000) DUMMY
      CCNTINUE
60 CONTINUE
61 IF(NPAR(14).EQ.0) NPAR(14)=1
      NEL=NPAR(14)-1
      READ(5,9002) INEL,IINC
9002 FCRMAT(15,30X,I5)
      IF(IINC.EQ.0) IINC=1
      READ(5,9004) (INP(I),I=1,8)
9004 READ(5,9004) (INP(I),I=9,N20)
      FCRMAT(1215)
      NEL=NEL+1
140 NREL(NEL)=NEL
      ML=INEL-NEL

```

```

00003020
00003030
00003040
00003050
00003060
00003070
00003080
00003090
00003100
00003110
00003120
00003130
00003140
00003150
00003160
00003170
00003180
00003190
00003200
00003210
00003220
00003230
00003240
00003250
00003260
00003270
00003280
00003290
00003300
00003310
00003320
00003330
00003340
00003350
00003360
00003370
00003380
00003390
00003400
00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490

```


00003500
00003510
00003520
00003530
00003540
00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720

```

150 IF (ML) 150, 155, 160
150 WRITE(6, 8000) NEL
8000 FORMAT(//5X, 'WARNING - ERROR IN ELEMENT ', I5//)
C *** NC GENERATION OF NODE POINTS REQUIRED
155 DO 156 I=1, N20
156 NP(I)=INP(I)
      NCONT(I, NEL)=NP(I)
      CONTINUE
156 GO TO 162
C *** GENERATION OF NODE POINTS REQUIRED
160 DO 161 I=1, N20
161 IF (NP(I).EQ.0) GO TO 161
      NP(I)=NP(I)+KN
      NCONT(I, NEL)=NP(I)
      CONTINUE
161 CONTINUE
162 NUMEL=NUMEL+1
      IF (NEL.EQ.NPAR(2)) RETURN
      IF (NEL.LT.IANEL) GO TO 143
      KN=IINC
      GC TO 130
900 GC TO 130 CONTINUE
      END
      SUBROUTINE TEMPRT(ZPT, T)
      IMPLICIT REAL*8(A-H, O-Z)
      COMMON/IPARA/NUMNP, NUMEL
      DIMENSION ZPT(1), T(271)
      EPS=1.D-4
      A=0.
      DO 1000 I=1, 2
      DO 100 I=1, NUMNP
      C1=ZPT(I)-3.385000
      C2=ZPT(I)-3.517500
      C3=ZPT(I)-3.650000
      C4=ZPT(I)-3.782500
      C5=ZPT(I)-3.915000
      C6=ZPT(I)-3.451250
      C7=ZPT(I)-3.5837500
      C8=ZPT(I)-3.7162500
      C9=ZPT(I)-3.8487500

```

```

CS=DABS(C9)
J=I+1
T(I)=A
IF(C1.LT.EPSI(J)=1850.
IF(C2.LT.EPSI(J)=1950.
IF(C3.LT.EPSI(J)=2000.
IF(C4.LT.EPSI(J)=2050.
IF(C5.LT.EPSI(J)=2065.
IF(C6.LT.EPSI(J)=1900.
IF(C7.LT.EPSI(J)=1975.
IF(C8.LT.EPSI(J)=2025.
IF(C9.LT.EPSI(J)=2060.
CCNTINUE
A=A+1.
WRITE(6,20) T
FCRMT(5X,5F20.5)
20 WRITE(56) T
1000 CONTINUE
ENDFILE 56
RETURN
END
/*
//GO.FT56F001 DD UNIT=3330,VOL=SER=DISK01,DSN=F0099.TAPE,DISP=OLD
//GO.SYSIN DD *
(PUT ADINA DECK HERE)
/*

```


INPUT DECK

103

[illegible][illegible]

RESULTS

ICPK1= 0
ICPK2= 1
ICPK3= 1
ICPK4= 1

TOTAL NUMBER OF NODE POINTS = 241
TOTAL NUMBER OF ELEMENTS = 24
THE NUMBER OF GAUSS POINTS = 4
THE NUMBER OF PRESSURE LOADS APPLIED = 6

THE VALUE OF MTOT IS 2258

PRESSURE LOAD VERIFICATION - BENDING SPECIMEN CASE 4

NEQ AND NCARD FOR IC IN GEOM1 = 642 0

NOOE	COF	NCUR	FACTOR
115	3	1	15625000000000006D-C1
118	3	1	-0.390625000000000048D-02
123	3	1	0.390625000000000044D-02
128	3	1	-0.15625000000000004D-01
131	3	1	-0.15625000000000004D-01
134	3	1	-0.31250000000000008D-01
137	3	1	-0.31250000000000008D-01
142	3	1	0.781250000000000103D-02
147	3	1	0.781250000000000103D-02
150	3	1	-0.15625000000000004D-01
153	3	1	-0.15625000000000004D-01
156	3	1	-0.31250000000000006D-C1
161	3	1	0.781250000000000103D-02
166	3	1	0.781250000000000105D-02
169	3	1	-0.15625000000000004D-01
172	3	1	-0.31250000000000003D-01
175	3	1	-0.31250000000000003D-01
180	3	1	0.781250000000000092D-02
185	3	1	0.781250000000000092D-02
188	3	1	-0.15625000000000005D-01
191	3	1	-0.15625000000000004D-01
194	3	1	-0.31250000000000009D-01
199	3	1	0.781250000000000103D-02
204	3	1	0.781250000000000103D-02
207	3	1	-0.15625000000000005D-01
210	3	1	-0.15625000000000004D-01
213	3	1	-0.31250000000000007D-01
218	3	1	0.781250000000000096D-02
223	3	1	0.781250000000000106D-02
226	3	1	-0.15625000000000003D-01
229	3	1	-0.15625000000000002D-01
232	3	1	-0.15624999999999999D-01
237	3	1	0.390625000000000058D-02
			0.390625000000000057D-C2

THE TOTAL NUMBER OF LOADS IS 33

APPENDIX D. LISTING OF DISTRIBUTED LOAD PREPROCESSOR

```

*****
IMPLICIT REAL*8(A-P,Q-Z)
*****
THIS PROGRAM IS A FINITE ELEMENT PREPROCESSOR THAT CAN
BE USED TO CALCULATE CONSISTENT PRESSURE LOADS.

1. CONTROL CARD (1015)
COLUMNS      VARIABLE
1-5           NUMNP
6-10          NUMEL
11-15         NGP
16-20         NCUR
21-25         NPLOAD
26-30         IPRINT1
31-35         IPRINT2
36-40         IPRINT3
41-45         IPRINT4

                MEANING
                TOTAL NUMBER OF NODES
                TOTAL NUMBER OF ELEMENTS
                NUMBER OF GAUSS POINTS
                USED IN INTEGRATION.(2-6)
                NUMBER OF ADINA LCAD CURVE
                THAT OUTPUT LOADS REFER TO.
                (USUALLY NCUR=1)
                NUMBER OF ELEMENT FACES WITH
                DISTRIBUTED LOADS.
                PRINT CONTROL
                PRINT CONTROL
                PRINT CONTROL
                PRINT CONTROL

2. NODAL PRESSURES CARD (215,8F10.5)
INPUT NPLOAD OF THESE CARDS
COLUMNS      VARIABLE
1-5           NPREL(J)
6-10          NFACE(J)
11-50         PRESS(I,J)

                MEANING
                ELEMENT NUMBER
                FACE TO WHICH PRESSURE
                APPLIED
                NODAL PRESSURE INTENSITIES
                INPUT HERE. CORNER NODES
*****

```

```

PRE00010
PRE00020
PRE00030
PRE00040
PRE00050
PRE00060
PRE00070
PRE00080
PRE00090
PRE00100
PRE00110
PRE00120
PRE00130
PRE00140
PRE00150
PRE00160
PRE00170
PRE00180
PRE00190
PRE00200
PRE00210
PRE00220
PRE00230
PRE00240
PRE00250
PRE00260
PRE00270
PRE00280
PRE00290
PRE00300
PRE00310
PRE00320
PRE00330
PRE00340
PRE00350
PRE00360
PRE00370
PRE00380
PRE00390
PRE00400
PRE00410
PRE00420
PRE00430
PRE00440
PRE00450
PRE00460
PRE00470
PRE00480

```

```

CCCCCCCCCCCCCCCCCCCC
      FIRST, THEN MIDSIDE NODES.
      POSITIVE PRESSURE WILL TEND
      TO COMPRESS ELEMENT.

      PRINT CONTROL
      ICHK1: IF (ICLK1.EQ.1) PROGRAM WILL JUST READ AND PRINT ADINA
              DECK. WILL NOT CALL SUBROUTINE TRANS.
      ICHK2: IF (ICLK2.EQ.1) PROGRAM WILL NOT PRINT ELEMENT-BY-
              ELEMENT RESULTS.
      ICHK3: IF (ICLK3.EQ.1) PROGRAM WILL NOT PUNCH DECK
      ICHK4: IF (ICLK4.EQ.1) PROGRAM WILL SKIP PRINTING INPUT INFORM-
              ATION. (WILL NOT CALL SUBROUTINE PRTOUT)

      JCHN PREISEL      NOVEMBER, 1977      NAVAL POSTGRADUATE SCHOOL
                        MONTEREY, CALIFORNIA 93940

*****
COMMON A(15000)
COMMON/IPARA/NUMNP, NUMEL, NGP, NCUR, NPLQAD
COMMON/CHECK/ICLK1, ICHK2, ICHK3, ICHK4
MTOT=15000
READ(5,1001) NUMNP, NUMEL, NGP, NCUR, NPLQAD, ICHK1, ICHK2, ICHK3,
1 ICHK4
WRITE(6,1002) ICHK1, ICHK2, ICHK3, ICHK4
FORMAT(//5X, 'ICLK1=', I5//5X, 'ICLK2=', I5//5X, 'ICLK3=', I5//
15X, 'ICLK4=', I5//)
FORMAT(10I5)
WRITE(6,1003) NUMNP, NUMEL, NGP, NPLQAD
FORMAT(//5X, 'TOTAL NUMBER OF NODE POINTS =', I5//5X, 'TOTAL NUMBER
1 OF ELEMENTS =', I5//5X, 'THE NUMBER OF GAUSS POINTS =', I5//5X, 'THE
2 NUMBER OF PRESSURE LOADS APPLIED =', I5//)
N1=1
N2=N1+3*NUMNP
N3=N2+NUMNP
N4=N3+NUMNP
N5=N4+NUMNP
N6=N5+NUMNP
N7=N6+NUMEL
N8=N7+20*NUMEL
N9=N8+8*NPLQAD

```

```

PRE00490
PRE00500
PRE00510
PRE00520
PRE00530
PRE00540
PRE00550
PRE00560
PRE00570
PRE00580
PRE00590
PRE00600
PRE00610
PRE00620
PRE00630
PRE00640
PRE00650
PRE00660
PRE00670
PRE00680
PRE00690
PRE00700
PRE00710
PRE00720
PRE00730
PRE00740
PRE00750
PRE00760
PRE00770
PRE00780
PRE00790
PRE00800
PRE00810
PRE00820
PRE00830
PRE00840
PRE00850
PRE00860
PRE00870
PRE00880
PRE00890
PRE00900
PRE00910
PRE00920
PRE00930
PRE00940
PRE00950
PRE00960

```

```

2001      N10=N9+NPLGAD
          NLAST=N10+2*NPLLOAD
          IF(NLAST.LT.MTOT)WRITE(6,2001)NLAST
          FORMAT(//5X,'THE VALUE OF MTOT IS',I7//)
2000      IF(NLAST.GT.MTOT)WRITE(6,2000)NLAST
          FORMAT(//5X,'MTOT SHOULD BE INCREASED TO',I7//)
          CALL CENTF(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),
            1A(N10))
          STOP
          END
          SUBROUTINE CENTF(TFVT,XPT,YPT,ZPT,NUMPT,NREL,NCONT,PRESS,NPREL,
            1NNFACE)
          C
          C*****
          C
          SUBROUTINE TO COORDINATE I/C FOR CCNSISTENT PRESSURE LOADING
          C*****
          C
          IMPLICIT REAL*8 (A-H,O-Z)
          COMMON/CCHK1,ICCHK2,ICCHK3,ICCHK4
          DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1)
          DIMENSION TFVT(3,1),PRESS(8,1),NCONT(20,1),NNFACE(1),NPREL(1)
          CALL RDADIN(XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)
          IF (ICCHK4.EQ.1)GO TO 10
          CALL PNTOUT(XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)
          IF (ICCHK1.EQ.1)RETURN
          CONTINUE
          CALL TRANS(TFVT,XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)
          RETURN
          END
          SUBROUTINE PNTOUT(XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)
          C
          C*****
          C
          SUBROUTINE TO PRINT OUT NODE AND ELEMENT INFORMATION
          C*****
          C
          IMPLICIT REAL*8 (A-H,O-Z)
          COMMON/IPARA/NUMNP,NUMEL,NGP,NCUR,NPLOAD
          DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCONT(20,1)
          DIMENSION PRESS(8,1),NPREL(1),NNFACE(1)
          C

```

```

PRE00970
PRE00980
PRE00990
PRE01000
PRE01010
PRE01020
PRE01030
PRE01040
PRE01050
PRE01060
PRE01070
PRE01080
PRE01090
PRE01100
PRE01110
PRE01120
PRE01130
PRE01140
PRE01150
PRE01160
PRE01170
PRE01180
PRE01190
PRE01200
PRE01210
PRE01220
PRE01230
PRE01240
PRE01250
PRE01260
PRE01270
PRE01280
PRE01290
PRE01300
PRE01310
PRE01320
PRE01330
PRE01340
PRE01350
PRE01360
PRE01370
PRE01380
PRE01390
PRE01400
PRE01410
PRE01420
PRE01430
PRE01440

```



```

DIMENSION IDOF(6),ID(6),IDOLD(6),NODE(20),NPAR(20)
1.NP(20),INP(20)
DATA CTEST/'X'
NCARD=0

C*****READ PRESSURE LOAD DATA*****
C
DO 9015 J=1,NPLOAD
FORMAT(2I5,8F5.0)
9014 READ(5,9014)NPREL(J),NNFACE(J),(PRESS(I,J),I=1,8)
9015 READ(5,9000)(TITLE(I),I=1,20)
WRITE(6,9011)(TITLE(I),I=1,20)
9011 FORMAT(//5X,20A4,///)
C *** READ MASTER CONTROL CARDS
C *** NUMNP = TOTAL NUMBER OF NODE POINTS
C *** NELTYP = NUMBER OF ELEMENT GROUPS
9001 READ(5,9001) NUMNP,(IDOF(I),I=1,6),NEGL,NEGNL,MODEX,NSTE
FORMAT(15,6I1,14,3I5)
NELTYP=NEGL+NEGNL
NNODE=NUMNP
READ(5,7002) IMASS,IDAMP,IMASSN,IDAMPN
7002 FORMAT(4I5)
READ(5,7002) IEIG
READ(5,7002) ISREF,NUMREF,IQUIT,ITEMAX
READ(5,9000) DUMMY
READ(5,9000) DUMMY
READ(5,9000) DUMMY
C *** READ OR GENERATE NODAL POINT DATA
NOLD=0
NEQ=0
9006 READ(5,9006)CT,N,(ID(I),I=1,6),XPT(N),YPT(N),ZPT(N),KN
9006 FORMAT(A1,14,1X,14,5I5,3F10.0,15)
C *** CHECK FOR CYLINDRICAL COORDINATES
IF(CT.NE.CTEST) GO TO 12
DUM=ZPT(N)/57.2958
R=YPT(N)
YPT(N)=R*DCOS(ZPT(N)/57.2958D0)
ZPT(N)=R*DSIN(ZPT(N)/57.2958D0)
12 CONTINUE
NUMPT(N)=N
IF(NOLD.EQ.0) GO TO 50
FOR GENERATION OF FIXED BOUNDARY CONDITIONS
DO 15 I=1,6
IF(IDOLD(I).EQ.-1.AND.ID(I).EQ.0) ID(I)=IDOLD(I)
CONTINUE
15 IF(KNOLD.EQ.0) GO TO 50
NUM=(N-NOLD)/KNOLD
NUMN=NUM-1

```

```

PRE01930
PRE01940
PRE01950
PRE01960
PRE01970
PRE01980
PRE01990
PRE02000
PRE02010
PRE02020
PRE02030
PRE02040
PRE02050
PRE02060
PRE02070
PRE02080
PRE02090
PRE02100
PRE02110
PRE02120
PRE02130
PRE02140
PRE02150
PRE02160
PRE02170
PRE02180
PRE02190
PRE02200
PRE02210
PRE02220
PRE02230
PRE02240
PRE02250
PRE02260
PRE02270
PRE02280
PRE02290
PRE02300
PRE02310
PRE02320
PRE02330
PRE02340
PRE02350
PRE02360
PRE02370
PRE02380
PRE02390
PRE02400

```



```

PRE02410
PRE02420
PRE02430
PRE02440
PRE02450
PRE02460
PRE02470
PRE02480
PRE02490
PRE02500
PRE02510
PRE02520
PRE02530
PRE02540
PRE02550
PRE02560
PRE02570
PRE02580
PRE02590
PRE02600
PRE02610
PRE02620
PRE02630
PRE02640
PRE02650
PRE02660
PRE02670
PRE02680
PRE02690
PRE02700
PRE02710
PRE02720
PRE02730
PRE02740
PRE02750
PRE02760
PRE02770
PRE02780
PRE02790
PRE02800
PRE02810
PRE02820
PRE02830
PRE02840
PRE02850
PRE02860
PRE02870
PRE02880

```

```

C *** IF (NUMN.LT.1) GO TO 50
      TO COUNT DOFS TO DETERMINE NUMBER CF IC CARDS
      DC 20 I=1,6
      IF (IDOF(I).EQ.0.AND.IDOLD(I).EQ.0) NEQ=NEQ+NUMN
20 CONTINUE
      DX=(XPT(N)-XPT(NOLD))/NUM
      IF (CT.NE.CTEST) GO TO 21
      ROLD=YPT(NOLD)/DCOS(DUMOLD)
      RNEW=YPT(N)/DCOS(DUM)
      OR=(RNEW-ROLD)/NUM
      DT=(DUM-DUMOLD)/NUM
      GO TO 22
21 CONTINUE
      DY=(YPT(N)-YPT(NOLD))/NUM
      DZ=(ZPT(N)-ZPT(NOLD))/NUM
22 CONTINUE
      K=NOLD
      DO 30 J=1,NUMN
      KK=K
      K=K+KNOLD
      XPT(K)=XPT(KK)+DX
      IF (CT.NE.CTEST) GO TO 26
      ROLD=ROLD+OR
      DUMOLD=DUMOLD+DT
      YPT(K)=ROLD*DCOS(DUMOLD)
      ZPT(K)=ROLD*DSIN(DUMOLD)
      GO TO 28
26 CONTINUE
      YPT(K)=YPT(KK)+DY
      ZPT(K)=ZPT(KK)+DZ
28 CONTINUE
      NUMPT(K)=K
30 CONTINUE
      NGLD=N
50 KNOLD=KN
      DUMOLD=DUM
      TO COUNT DOFS TO DETERMINE NUMBER CF IC CARDS
      DC 55 I=1,6
      IF (IDOF(I).EQ.0.AND.ID(I).EQ.0) NEQ=NEQ+1
      IDOLD(I)=ID(I)
55 CONTINUE
      IF (N.NE.NUMNP) GO TO 10
      READ LOAD CONTROL CARDS
      READ(5,9012)NLOAD,NLCUR,NPTM,IDGRAV
9012 FORMAT(4I5)
      DC 80 I=1,IMASSN
      IF (IMASSN.EQ.0) GO TO 81
      READ(5,9000) DUMMY

```

```

80 CONTINUE
81 IF (IDAMPN.EQ.0) GO TO 91
DC 90 I=1, ICAMPN
READ (5,9000) DUMMY
CONTINUE
90 CONTINUE
91 READ INITIAL CONDITIONS
READ (5,9002) ICON
IF (ICON.EQ.0) GO TO 100
CARDNR=NEQ/6.0
TEST=CARDNR-NCARD
IF (TEST.GT.0.1) NCARD=NCARD+1
DC 95 I=1, NCARD
READ (5,9000) DUMMY
CONTINUE
95 IF (IMASS.EQ.0) GO TO 100
DC 96 I=1, NCARD
READ (5,9000) DUMMY
CONTINUE
96 DC 98 I=1, NCARD
READ (5,9000) DUMMY
CONTINUE
98 FORMAT (6E12.6)
9007 CONTINUE
100 NUMEL=0
WRITE (6,9009) NEQ, NCARD
FORMAT (///, 'NEQ AND NCARD FOR IC IN GEOM1 = ', I5, 10X, I5, ///)
C ***
9009 READ ELEMENT CONTROL CARDS
DC 900 M=1, NELTYP
READ (5,9008) (NPAR(I), I=1, 20)
WRITE (6,9010) (NPAR(I), I=1, 20)
FORMAT (20I4)
9008 CONTINUE
9010 FORMAT (///, 'NPAR = ', 20I5, ///)
MTYPE=NPAR(1)
NUMMAT=NPAR(16)
NSTRES=NPAR(13)
CALCULATE THE NUMBER OF MATERIAL CASE CARDS
IF (NPAR(15).EQ.1) NCARD=1
IF (NPAR(15).EQ.2) NCARD=2+NPAR(18)
IF (NPAR(15).EQ.3) NCARD=4
IF (NPAR(15).EQ.4) NCARD=4
IF (NPAR(15).EQ.5) NCARD=2
IF (NPAR(15).EQ.8) NCARD=1
IF (NPAR(15).EQ.9) NCARD=1
IF (NPAR(15).EQ.10) NCARD=6
IF (NPAR(15).EQ.11) NCARD=6

```

```

PRE02890
PRE02900
PRE02910
PRE02920
PRE02930
PRE02940
PRE02950
PRE02960
PRE02970
PRE02980
PRE02990
PRE03000
PRE03010
PRE03020
PRE03030
PRE03040
PRE03050
PRE03060
PRE03070
PRE03080
PRE03090
PRE03100
PRE03110
PRE03120
PRE03130
PRE03140
PRE03150
PRE03160
PRE03170
PRE03180
PRE03190
PRE03200
PRE03210
PRE03220
PRE03230
PRE03240
PRE03250
PRE03260
PRE03270
PRE03280
PRE03290
PRE03300
PRE03310
PRE03320
PRE03330
PRE03340
PRE03350
PRE03360

```

```

PRE03370
PRE03380
PRE03390
PRE03400
PRE03410
PRE03420
PRE03430
PRE03440
PRE03450
PRE03460
PRE03470
PRE03480
PRE03490
PRE03500
PRE03510
PRE03520
PRE03530
PRE03540
PRE03550
PRE03560
PRE03570
PRE03580
PRE03590
PRE03600
PRE03610
PRE03620
PRE03630
PRE03640
PRE03650
PRE03660
PRE03670
PRE03680
PRE03690
PRE03700
PRE03710
PRE03720
PRE03730
PRE03740
PRE03750
PRE03760
PRE03770
PRE03780
PRE03790
PRE03800
PRE03810
PRE03820
PRE03830
PRE03840

```

```

IF(NPAR(15).NE.12) GO TO 111
CARDNR=NPAK(17)/8.0
NCARD=IDINT(CARDNR)
TEST=CARDNR-NCARD
IF(TEST.GT.0.1) NCARD=NCARD+1
111 CONTINUE
N2J=2J
C *** READ MATERIAL PROPERTIES
DC 222 J=1,NUMMAT
9000 FORMAT(20A4)
DO 45 I=1,NCARD
READ(5,9000) DUMMY
CONTINUE
45
222 CCNTINUE
READ STRESS OUTPUT TABLE CARDS
C *** IF(NPAR(13).EQ.0) GO TO 61
DO 60 I=1,NSTRES
READ(5,9000) DUMMY
CONTINUE
60
61 CCNTINUE
IF(NPAR(14).EQ.0) NPAR(14)=1
NEL=NPAR(14)-1
INEL,IINC
130 FORMAT(15,30X,15)
9002 IF(IINC.EQ.0) IINC=1
READ(5,9004) (INP(I),I=1,8)
9004 READ(5,9004) (INP(I),I=9,N2J)
140 FORMAT(12I5)
NEL=NEL+1
NREL(NEL)=NEL
ML=INEL-NEL
IF(ML) 150,155,160
150 WRITE(6,8030)INEL
8000 FORMAT(//5X,'WARNING - ERROR IN ELEMENT ',15//)
C *** NO GENERATION OF NODE POINTS REQUIRED
155 DC 156 I=1,N20
NP(I)=INP(I)
NCONT(I,NEL)=NP(I)
156 CONTINUE
GO TO 162
C *** GENERATION OF NODE POINTS REQUIRED
160 DO 161 I=1,N20
IF(NP(I).EQ.0) GO TO 161
NP(I)=NP(I)+KN
NCONT(I,NEL)=NP(I)
161 CONTINUE
162 CONTINUE

```



```

30      CCD(I,3)=ZPT(K)
      CONTINUE
C
C      REAC PRESSURES INPUT AT UP TO 8 NODES,CORNER NODES FIRST
C      THEN MIDSIDE NODES
C
35      DO 35 I=1,8
      PWA(I)=PRESS(I,KOUNT)
C
C      ECHO PRINT ALL INPUT DATA
C
      IF(ICHK2.EQ.1)GO TO 105
      WRITE(6,190)
      FORMAT(//15X,'MATRIX OF NATURAL COORDINATES'////)
      CALL PRINTD(COD,20,3,2,TIT,20)
      WRITE(6,191)
      FORMAT(//10X,'MATRIX OF CONNECTIVITIES'///)
      WRITE(6,195)NCON
      FORMAT(//5X,2015////)
      WRITE(6,192)
      FORMAT(//10X,'MATRIX OF NODAL PRESSURES'///)
      CALL PRINTD(PWA,1,8,2,TIT,1)
      WRITE(6,100)IEL,NGP,NFACE
      FORMAT(//10X,'THE ELEMENT NUMBER IS ',I4//' THE NUMBER OF
      ITS IS ',I3//' THE FACE TO WHICH PRESSURE IS APPLIED IS ',I3//)
      CCNTINUE
      CALL CUBAT(NGP)
      IF(ICHK2.EQ.1)GO TO 207
      CALL PRINTD(FVT,1,60,1,TIT,1)
      SUMX=0.000
      SUMY=0.000
      SUMZ=0.000
      DO 201 I=1,20
      SUMX=SUMX+FVT(I)
      DO 202 I=21,40
      SUMY=SUMY+FVT(I)
      DO 203 I=41,60
      SUMZ=SUMZ+FVT(I)
      WRITE(6,204)SUMX,SUMY,SUMZ
      FORMAT(//10X,'SUMX =',G25.16//' SUMY =',G25.16//' SUMZ =',G25.16//)
      CCNTINUE
      DC 205 I=1,8
      KK=NODES(I)
      KK=NCON(K)
      TFVT(1,KK)=TFVT(1,KK)+FVT(K)
      TFVT(2,KK)=TFVT(2,KK)+FVT(K+20)

```

```

PRE04330
PRE04340
PRE04350
PRE04360
PRE04370
PRE04380
PRE04390
PRE04400
PRE04410
PRE04420
PRE04430
PRE04440
PRE04450
PRE04460
PRE04470
PRE04480
PRE04490
PRE04500
PRE04510
PRE04520
PRE04530
PRE04540
PRE04550
PRE04560
PRE04570
PRE04580
PRE04590
PRE04600
PRE04610
PRE04620
PRE04630
PRE04640
PRE04650
PRE04660
PRE04670
PRE04680
PRE04690
PRE04700
PRE04710
PRE04720
PRE04730
PRE04740
PRE04750
PRE04760
PRE04770
PRE04780
PRE04790
PRE04800

```

```

205 TFVT(3, KK1)=TFVT(3, KK)+FVT(K+40)
206 CONTINUE
209 WRITE(6, 209)
    FORMAT(//5X, 'NODE', 10X, 'X COMPONENT', 12X, 'Y COMPONENT', 15X,
    1, 2 COMPONENT', //)
    EPS=1.D-8
    DO 210 NODE=1, NUMNP
        A=DABS(TFVT(1, NODE))
        B=DABS(TFVT(2, NODE))
        C=DABS(TFVT(3, NODE))
        IF(A.GT.EPS.OR.B.GT.EPS.OR.C.GT.EPS)WRITE(6, 211)NODE,
        1(TFVT(1, NODE), I=1, 3)
        FC=FORMAT(1X, 15, 4X, 3G25.16)
        CONTINUE
        WRITE(6, 213)
        FC=FORMAT(//1X, 'NODE', 2X, 'DOF', 1X, 'NCUR', 12X, 'FACTOR', //)
        IOUT=6
        KLCAD=0
        DO 215 NODE=1, NUMNP
            A=DABS(TFVT(1, NODE))
            IF(A.GT.EPS)KLOAD=KLOAD+1
            AA=TFVT(1, NODE)
            IF(A.GT.EPS)WRITE(IOUT, 216)NODE, NCUR, AA
            B=DABS(TFVT(2, NODE))
            IF(B.GT.EPS)KLOAD=KLOAD+1
            BB=TFVT(2, NODE)
            IF(B.GT.EPS)WRITE(IOUT, 217)NODE, NCUR, BB
            C=DABS(TFVT(3, NODE))
            IF(C.GT.EPS)KLOAD=KLOAD+1
            CC=TFVT(3, NODE)
            IF(C.GT.EPS)WRITE(IOUT, 218)NODE, NCUR, CC
        CONTINUE
        EQ=6)WRITE(6, 250)KLOAD
        FC=FORMAT(//5X, 'THE TOTAL NUMBER OF LOADS IS ', 15//)
        FC=FORMAT(15, 4X, 1, 15, G25.16)
        FC=FORMAT(15, 4X, 2, 15, G25.16)
        FC=FORMAT(15, 4X, 3, 15, G25.16)
        IF(IOUT.EQ.7)STOP
        IF(1CHK3.NE.1)IOUT=7
        IF(1CHK3.NE.1)GO TO 214
        STOP
        END
        SUBROUTINE CUBAT(NGP)
            C
            C
            PROF GILLES CANTIN AT THE NAVAL POSTGRADUATE SCHOOL SEPT. 1977
            IMPLICIT REAL*8(A-H,O-Z)

```

```

PRE04810
PRE04820
PRE04830
PRE04840
PRE04850
PRE04860
PRE04870
PRE04880
PRE04890
PRE04900
PRE04910
PRE04920
PRE04930
PRE04940
PRE04950
PRE04960
PRE04970
PRE04980
PRE04990
PRE05000
PRE05010
PRE05020
PRE05030
PRE05040
PRE05050
PRE05060
PRE05070
PRE05080
PRE05090
PRE05100
PRE05110
PRE05120
PRE05130
PRE05140
PRE05150
PRE05160
PRE05170
PRE05180
PRE05190
PRE05200
PRE05210
PRE05220
PRE05230
PRE05240
PRE05250
PRE05260
PRE05270
PRE05280

```



```

IF(NFACE.EQ.6)T=-1.0D0
IF(NFACE.EQ.5.OR.NFACE.EQ.6)GO TO 140
T = GP(K,IDGP)
WT = WF(K,IDGP)
CONTINUE

```

140

```

WFAC = WR*WS*WT
IF(CHK2.EQ.1)GO TO 145
WRITE(6,160)R,S,T,WFAC
CONTINUE

```

145

```

CALL SHP20(R,S,T)
CONTINUE
FORMAT(1X,4G25.16)
RETURN
END

```

200

160

```

SUBROUTINE SHP20(R,S,T)

```

CCCCCCCCCCCCCCCCCCCC

UPON EXIT FROM THE PRESENT SUBROUTINE, FVT WILL CONTAIN
THE VALUES OF LOAD VECTOR ARGUMENT FVT(20,3) THAT CAN BE
USED FOR GAUSSIAN QUADRATURE

THE MATRIX ARGUMENT IS THE RESULT OF THE PRODUCT OF THE
FOLLOWING MATRICES: NT(20,3)*A(3,3)*N(3,20)*XYZ(20,3)*DETJ
WHERE THE MATRIX N(3,20) IS THE MATRIX OF SHAPE FUNCTIONS
NT(20,3) IS THE TRANSPOSE OF N(3,20), XYZ(20,3) IS THE
MATRIX OF COORDINATES OF THE NODES AND DETJ IS THE DETERMINANT
OF THE JACOBIAN MATRIX

R,S,T ARE THE NON DIMENSIONAL COORDINATES OF THE ELEMENT
DETJ IS THE DETERMINANT OF THE JACOBIAN MATRIX
FVT IS THE OUTPUT VECTOR
COD IS AN INPUT ARRAY OF THE ELEMENT COORDINATES IN THE
GLOBAL COORDINATE SYSTEM
NCON IS THE CONNECTIVITY MATRIX OF THE ELEMENT
IEL IS THE ELEMENT IDENTIFICATION NUMBER

PROF. GILLES CANTIN SEPT. 1977 NAVAL POSTGRADUATE SCHCCL
MONTEREY CALIFORNIA 93940

```

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION CCD(20,3),CNI(3,20),EMI(20)
DIMENSION NCON(20)
REAL*8 JAC(3,3)
COMMON/GEOM/COD,NCON,IEL
GP(BT) = 0.5D0*(1.0D0 + BT)
GM(BT) = 0.5D0*(1.0D0 - BT)
GZ(BT) = 1.0D0 - BT*BT
DP(BT) = 0.5D0+ZRO*BT
DM(BT) = -0.5D0+ZRO*BT

```

PRE05770
PRE05780
PRE05790
PRE05800
PRE05810
PRE05820
PRE05830
PRE05840
PRE05850
PRE05860
PRE05870
PRE05880
PRE05890
PRE05900
PRE05910
PRE05920
PRE05930
PRE05940
PRE05950
PRE05960
PRE05970
PRE05980
PRE05990
PRE06000
PRE06010
PRE06020
PRE06030
PRE06040
PRE06050
PRE06060
PRE06070
PRE06080
PRE06090
PRE06100
PRE06110
PRE06120
PRE06130
PRE06140
PRE06150
PRE06160
PRE06170
PRE06180
PRE06190
PRE06200
PRE06210
PRE06220
PRE06230
PRE06240

520	DNI(1,1,1) =	DZ(R)*GM(S)*GP(T)	PRE06730
	DNI(2,1,1) =	GZ(R)*GM(S)*GP(T)	PRE06740
	DNI(3,1,1) =	GZ(R)*GM(S)*DP(T)	PRE06750
	IF(NCON(12).EQ.0)	GO TO 530	PRE06760
	DNI(1,1,2) =	GP(R)*DZ(S)*GP(T)	PRE06770
	DNI(2,1,2) =	GP(R)*DZ(S)*GP(T)	PRE06780
	DNI(3,1,2) =	GP(R)*DZ(S)*DP(T)	PRE06790
530	IF(NCON(13).EQ.0)	GO TO 540	PRE06800
	DNI(1,1,3) =	DZ(R)*GP(S)*GM(T)	PRE06810
	DNI(2,1,3) =	GZ(R)*DP(S)*GM(T)	PRE06820
	DNI(3,1,3) =	GZ(R)*GP(S)*DM(T)	PRE06830
540	IF(NCON(14).EQ.0)	GO TO 550	PRE06840
	DNI(1,1,4) =	DM(R)*GZ(S)*GM(T)	PRE06850
	DNI(2,1,4) =	GM(R)*DZ(S)*DM(T)	PRE06860
	DNI(3,1,4) =	GM(R)*GZ(S)*DM(T)	PRE06870
550	IF(NCON(15).EQ.0)	GO TO 560	PRE06880
	DNI(1,1,5) =	DZ(R)*GM(S)*GM(T)	PRE06890
	DNI(2,1,5) =	GZ(R)*DM(S)*GM(T)	PRE06900
	DNI(3,1,5) =	GZ(R)*GM(S)*DM(T)	PRE06910
560	IF(NCON(16).EQ.0)	GO TO 570	PRE06920
	DNI(1,1,6) =	GP(R)*GZ(S)*GM(T)	PRE06930
	DNI(2,1,6) =	GP(R)*DZ(S)*GM(T)	PRE06940
	DNI(3,1,6) =	GP(R)*GZ(S)*DM(T)	PRE06950
570	IF(NCON(17).EQ.0)	GO TO 580	PRE06960
	DNI(1,1,7) =	GP(R)*GP(S)*GZ(T)	PRE06970
	DNI(2,1,7) =	GP(R)*DP(S)*GZ(T)	PRE06980
	DNI(3,1,7) =	GP(R)*GP(S)*DZ(T)	PRE06990
580	IF(NCON(18).EQ.0)	GO TO 590	PRE07000
	DNI(1,1,8) =	GP(S)*GZ(T)	PRE07010
	DNI(2,1,8) =	DM(R)*DP(S)*GZ(T)	PRE07020
	DNI(3,1,8) =	GM(R)*GP(S)*DZ(T)	PRE07030
590	IF(NCON(19).EQ.0)	GO TO 600	PRE07040
	DNI(1,1,9) =	DM(R)*GM(S)*GZ(T)	PRE07050
	DNI(2,1,9) =	GM(R)*DM(S)*GZ(T)	PRE07060
	DNI(3,1,9) =	GM(R)*GM(S)*DZ(T)	PRE07070
600	IF(NCON(20).EQ.0)	GO TO 620	PRE07080
	DNI(1,1,20) =	GP(R)*GM(S)*GZ(T)	PRE07090
	DNI(2,1,20) =	GP(R)*DM(S)*GZ(T)	PRE07100
620	DNI(3,1,20) =	GP(R)*GM(S)*DZ(T)	PRE07110
	CONTINUE		PRE07120
	DNI(1,1) =	DP(R)*GP(S)*GP(T)	(DNI(1, 9)+DNI(1,12)+DNI(1,17))/TWOP
	DNI(2,1) =	GP(R)*DP(S)*GP(T)	(DNI(2, 9)+DNI(2,12)+DNI(2,17))/TWOP
	DNI(3,1) =	GP(R)*GP(S)*DP(T)	(DNI(3, 9)+DNI(3,12)+DNI(3,17))/TWOP
	DNI(1,2) =	DM(R)*GP(S)*GP(T)	(DNI(1, 9)+DNI(1,10)+DNI(1,18))/TWOP
	DNI(2,2) =	GM(R)*DP(S)*GP(T)	(DNI(2, 9)+DNI(2,10)+DNI(2,18))/TWOP
	DNI(3,2) =	GM(R)*GP(S)*DP(T)	(DNI(3, 9)+DNI(3,10)+DNI(3,18))/TWOP
	DNI(1,3) =	DM(R)*GM(S)*GP(T)	(DNI(1, 9)+DNI(1,11)+DNI(1,19))/TWOP
	DNI(2,3) =	GM(R)*DM(S)*GP(T)	(DNI(2, 9)+DNI(2,11)+DNI(2,19))/TWOP
	DNI(3,3) =	GM(R)*GM(S)*GP(T)	(DNI(3, 9)+DNI(3,11)+DNI(3,19))/TWOP

```

DNI(3,3) = GM(R)*GM(S)*DP(T) - (DNI(3,13)+DNI(1,11)+DNI(3,11)+DNI(3,19))/TWCPR07210
DNI(1,4) = DP(R)*GM(S)*GP(T) - (DNI(1,11)+DNI(1,12)+DNI(2,12)+DNI(1,20))/TWCPR07220
DNI(2,4) = GP(R)*DM(S)*GP(T) - (DNI(2,11)+DNI(2,12)+DNI(3,12)+DNI(2,20))/TWCPR07230
DNI(3,4) = GP(R)*GM(S)*DP(T) - (DNI(3,11)+DNI(3,12)+DNI(3,20))/TWCPR07240
DNI(1,5) = DP(R)*GP(S)*GM(T) - (DNI(1,13)+DNI(1,16)+DNI(2,16)+DNI(1,17))/TWCPR07250
DNI(2,5) = GP(R)*GP(S)*GM(T) - (DNI(2,13)+DNI(2,16)+DNI(3,16)+DNI(2,17))/TWCPR07260
DNI(3,5) = GP(R)*GP(S)*DM(T) - (DNI(3,13)+DNI(3,16)+DNI(3,17))/TWCPR07270
DNI(1,6) = DM(R)*DP(S)*GM(T) - (DNI(1,13)+DNI(1,14)+DNI(2,14)+DNI(1,18))/TWCPR07280
DNI(2,6) = GP(R)*DP(S)*GM(T) - (DNI(2,13)+DNI(2,14)+DNI(3,14)+DNI(2,18))/TWCPR07290
DNI(3,6) = DM(R)*GP(S)*GM(T) - (DNI(3,13)+DNI(3,14)+DNI(3,18))/TWCPR07300
DNI(1,7) = DM(R)*GM(S)*GM(T) - (DNI(1,14)+DNI(1,15)+DNI(2,15)+DNI(1,19))/TWCPR07310
DNI(2,7) = DM(R)*DM(S)*GM(T) - (DNI(2,14)+DNI(2,15)+DNI(3,15)+DNI(2,19))/TWCPR07320
DNI(3,7) = GM(R)*GM(S)*DM(T) - (DNI(3,14)+DNI(3,15)+DNI(3,19))/TWCPR07330
DNI(1,8) = DP(R)*GM(S)*GM(T) - (DNI(1,15)+DNI(1,16)+DNI(2,16)+DNI(1,20))/TWCPR07340
DNI(2,8) = DP(R)*DM(S)*GM(T) - (DNI(2,15)+DNI(2,16)+DNI(3,16)+DNI(2,20))/TWCPR07350
DNI(3,8) = GP(R)*GM(S)*DM(T) - (DNI(3,15)+DNI(3,16)+DNI(3,20))/TWCPR07360
DC 630 I=1,3
DC 630 J=1,3
JAC(I,J)=ZRC
DC 630 K=1,20
630 JAC(I,J)=JAC(I,J)+DNI(I,K)*COC(K,J)
A1 = JAC(1,1)*JAC(2,2)*JAC(3,3)
A2 = JAC(1,2)*JAC(2,3)*JAC(3,1)
A3 = JAC(1,3)*JAC(2,1)*JAC(3,2)
B1 = JAC(3,1)*JAC(2,2)*JAC(1,1)
B2 = JAC(3,2)*JAC(2,3)*JAC(1,1)
B3 = JAC(3,3)*JAC(2,1)*JAC(1,2)
DTJ = A1+A2+A3 - (B1+B2+B3)
CALL FACEP(EMI,DNI,JAC)
IF(DTJ.LE.ZRO) GO TO 960
RETURN
960 WRITE(6,1000)IEL,DTJ,R,S,T
1000 FORMAT(/,5X,'ELEMENT NUMBER',15,' HAS A JACOBIAN DETERMINANT',/
1,5X,'OF VALUE',16,'** EXECUTION STOPS **',/
2,5X,' R =',625.16/5X,' S =',625.16/5X,' T =',625.16//)
STOP
END
SUBROUTINE FACEP(EMI,DNI,JAC)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION NCODES(8),KFACE(6,8),FVAL(6),PR(8),P(20)
DIMENSION PHA(8),EMI(20),DNI(3,20),DUM(60),FVT(60)
REAL*8 JAC(3,3),NORML(3)
COMMON/PRESR/PWA,MFAC,NFACE
COMMON/OUTPUT/FVT,NODES
DATA KFACE / 1,2,1,4,1,5,
4,3,5,8,2,6,
8,7,6,7,3,7,
1
2

```



```

PRE07690
PRE07700
PRE07710
PRE07720
PRE07730
PRE07740
PRE07750
PRE07760
PRE07770
PRE07780
PRE07790
PRE07800
PRE07810
PRE07820
PRE07830
PRE07840
PRE07850
PRE07860
PRE07870
PRE07880
PRE07890
PRE07900
PRE07910
PRE07920
PRE07930
PRE07940
PRE07950
PRE07960
PRE07970
PRE07980
PRE07990
PRE08000
PRE08010
PRE08020
PRE08030
PRE08040
PRE08050
PRE08060
PRE08070
PRE08080
PRE08090
PRE08100
PRE08110
PRE08120
PRE08130
PRE08140
PRE08150
PRE08160

```

```

3      5,6,12,3,3,4,8,
4      12,10,17,20,9,13,
5      20,19,13,15,13,14,
6      16,14,18,19,11,15,
7      17,18,9,11,12,16/
DATA FVAL /1.0D0,-1.0D0,1.0D0,-1.0D0,-1.0D0,-1.0D0/
DETERMINE ELEMENT NODES CONTRIBUTING TO FORCE CALCULATIONS
ON THIS FACE
DO 2 I=1,8
  NODES(I)=KFACE(NFACE,I)
CONTINUE
SET UP PRESSURE VECTOR FOR UP TO EIGHT FACE NODES
1. ADJUST THE SIGN OF THE PRESSURES SO THE POSITIVE
   PRESSURES ALWAYS TENSION THE ELEMENT
FACT=-FVAL(NFACE)
2. DISTRIBUTED PRESSURES GIVEN AT UP TO EIGHT NODES
   ON THE FACE
DC 25 K=1,8
PWA(I) = VALUE OF PRESSURE AT THE ITH NODE
PR(K)=PWA(K)*FACT
CONTINUE
CCOMPUTE THE DIRECTION NUMBERS: NORML(1),NORML(2),NORML(3)
IF(NFACE.GT.2)GO TO 60
NORML(1)=JAC(2,2)*JAC(3,3)-JAC(2,3)*JAC(3,2)
NORML(2)=JAC(2,3)*JAC(3,1)-JAC(2,1)*JAC(3,3)
NORML(3)=JAC(2,1)*JAC(3,2)-JAC(2,2)*JAC(3,1)
GO TO 94
IF(NFACE.GT.4)GO TO 80
NORML(1)=JAC(3,2)*JAC(1,3)-JAC(3,3)*JAC(1,2)
NORML(2)=JAC(3,3)*JAC(1,1)-JAC(3,1)*JAC(1,3)
NORML(3)=JAC(3,1)*JAC(1,2)-JAC(3,2)*JAC(1,1)
GO TO 94
CONTINUE
NORML(1)=JAC(1,2)*JAC(2,3)-JAC(1,3)*JAC(2,2)
NORML(2)=JAC(1,3)*JAC(2,1)-JAC(1,1)*JAC(2,3)
NORML(3)=JAC(1,1)*JAC(2,2)-JAC(1,2)*JAC(2,1)
CONTINUE
IF(NFACE.EQ.1.OR.NFACE.EQ.3.OR.NFACE.EQ.5)GO TO 100

```



```

95      CC 95 I=1,3
100     NORML(I)=-1.000*NORML(I)
C      CONTINUE
C
C      ASSEMBLE LARGE SHAPE FACTOR MATRIX: BIGN(3,60)
C
C      ZERO ALL ELEMENTS OF BIGN
C      DO 105 I=1,3
C      DC 110 J=1,60
C      BIGN(I,J)=0.000
C      CONTINUE
C      DO 120 J=1,20
C      BIGN(1,J)=EMI(J)
C      DC 130 J=21,40
C      K=J-20
C      BIGN(2,J)=EMI(K)
C      DC 140 J=41,60
C      K=J-40
C      BIGN(3,J)=EMI(K)
C
C      ASSEMBLE PRESSURE MATRIX(3,20)
C
C      ZERO ALL ELEMENTS OF P MATRIX
C
C      DO 150 I=1,20
C      P(I)=C.000
C      DO 160 I=1,8
C      J=NODES(I)
C      P(J)=PWA(I)
C
C      PERFORM MATRIX MULTIPLICATION: NORML(1,3)*BIGN(3,60)*NT(1,20)
C      *P(20,1)
C      PROD1(1,60)=NORML*BIGN; PROD2(1)=NT*P
C
C      DO 170 J=1,60
C      PROD1(J)=0.000
C      DO 170 K=1,3
C      PROD1(J)=PROD1(J)+NORML(K)*BIGN(K,J)
C      PROD2=J.000
C      DO 180 I=1,20
C      PROD2=PROD2+EMI(I)*P(I)
C
C      PERFORM MULTIPLICATION OF PROD1*PROD2
C
C      DO 190 J=1,60
C      DUM(J)=PROD1(J)*PROD2
C

```

```

PRE08170
PRE08180
PRE08190
PRE08200
PRE08210
PRE08220
PRE08230
PRE08240
PRE08250
PRE08260
PRE08270
PRE08280
PRE08290
PRE08300
PRE08310
PRE08320
PRE08330
PRE08340
PRE08350
PRE08360
PRE08370
PRE08380
PRE08390
PRE08400
PRE08410
PRE08420
PRE08430
PRE08440
PRE08450
PRE08460
PRE08470
PRE08480
PRE08490
PRE08500
PRE08510
PRE08520
PRE08530
PRE08540
PRE08550
PRE08560
PRE08570
PRE08580
PRE08590
PRE08600
PRE08610
PRE08620
PRE08630
PRE08640

```



```

      FM2(2)=F(4,ICDDE)
      NPR = 1 + (M-1)/NK
      N1 = 1
      N2 = MINO(M,NK)
      WRITE(6,1000) TITLE
      FFORMAT(//10A8/)
1000 DO 200 K=1,NPR
      WRITE(6,FM1) (J,J=N1,N2)
      DO 100 I=1,N
1100 WRITE(6,FM2) I,(A(I,J),J=N1,N2)
      N1 = N1 + NK
      N2 = MINO(M,N2+NK)
      RETURN
      ENTRY PRINTD(B,N,M,ICODE,TITLE,ND)
      DIMENSION B(ND,1)
      REAL*8 B
      IF(ICODE.LT.1.OR.ICODE.GT.5) ICODE=4
      NK=NCOL(ICODE)
      FM1(1)=F(1,ICODE)
      FM1(2)=F(2,ICODE)
      FM2(1)=F(3,ICODE)
      FM2(2)=F(4,ICODE)
      NPR = 1 + (M-1)/NK
      N1 = 1
      N2 = MINO(M,NK)
      WRITE(6,1000) TITLE
      DO 210 K=1,NPR
      WRITE(6,FM1) (J,J=N1,N2)
      DO 110 I=1,N
110 WRITE(6,FM2) I,(B(I,J),J=N1,N2)
      N1 = N1 + NK
      N2 = MINO(M,N2+NK)
      RETURN
      END

```

```

PRE09130
PRE09140
PRE09150
PRE09160
PRE09170
PRE09180
PRE09190
PRE09200
PRE09210
PRE09220
PRE09230
PRE09240
PRE09250
PRE09260
PRE09270
PRE09280
PRE09290
PRE09300
PRE09310
PRE09320
PRE09330
PRE09340
PRE09350
PRE09360
PRE09370
PRE09380
PRE09390
PRE09400
PRE09410
PRE09420
PRE09430
PRE09440
PRE09450
PRE09460

```


LIST OF REFERENCES

1. Admiralty Materials Laboratory Report No. 1/75, The Properties of Silicon Nitride Ceramics and their Use in Service Thermal Environments, by D. J. Godfrey, February 1975.
2. Tree, D. J. et al, "Three Dimensional Optimization of a Gas Turbine Disk and Blade Attachment", Journal of Aircraft, Vol. 13, p. 994-999, December 1976.
3. AiResearch Manufacturing Company of Arizona Report 76-212409, Documentation of Ceramic Component Design Analyses - Ceramic Gas Turbine Engine Demonstration Program, Appendix II, 1 December 1976.
4. Massachusetts Institute of Technology Report 82448-1, A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis, by K. J. Bathe, September 1975.
5. Earthquake Engineering Research Center Report EERC 73-11, SAP IV - A Structural Analysis Program for Static and Dynamic Response of Linear Systems, by K. J. Bathe, E. L. Wilson, F. E. Peterson, June 1973.
6. AiResearch Manufacturing Company of Arizona Report 76-212188(3), Ceramic Gas Turbine Engine Demonstration Interim Report Number 3, p. 24-30, December 1976.
7. Roark, R. J. and Young, W. C., Formulas for Stress and Strain, Fifth ed., Table 3, McGraw-Hill Publishing Company, 1975.
8. Bathe, K. J. and Wilson, E. L., Numerical Methods in Finite Element Analysis, Prentice Hall Inc., 1976.
9. Boley, B. A., and Weiner, J. H., Theory of Thermal Stresses, John Wiley and Sons, Inc., 1960.
10. AiResearch Manufacturing Company of Arizona, TL 310-1936, Sheets 1 through 10, Blade Sections Rotor First Stage Turbine, 20 September 1976.
11. Kibler, A. E., A Finite Element Preprocessor for SAP IV and ADINA, M.S. Thesis, Naval Postgraduate School, Monterey, CA., September 1977.

12. Easterling, L. R., Stress Analysis of Ceramic Gas Turbine Blades by the Finite Element Method - Part I, M.E. Thesis, Naval Postgraduate School, Monterey, CA, March 1978.
13. AiResearch Manufacturing Company of Arizona Report 76-212188(2), Ceramic Gas Turbine Engine Demonstration Interim Report Number 3, p. 2.5 - 7, September 1976.
14. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill Publishing Company, 1971.
15. W. R. Church Computer Center, Naval Postgraduate School, Monterey, CA, User's Manual, March 1970.
16. Losh, D. M., The Implementation of a Finite Element Computer Code and Associated Pre- and Postprocessor into AE 4101 and 4102 (Flight Vehicle Structural Analysis I and II), M.S. Thesis, Naval Postgraduate School, Monterey, CA, December 1976.
17. Naval Postgraduate School Technical Note TN No. 0141-05, User Libraries and Source Code Editing Under OS, by Sharon D. Raney, Revised July 1977.
18. Timoshenko, S. and Goodier, J. N., Theory of Elasticity, p. 254, 2nd ed., McGraw-Hill Book Company, 1951.